

日本国特許庁
JAPAN PATENT OFFICE

19.4.2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2004年 3月12日

RECD 10 JUN 2004

出願番号
Application Number: 特願2004-071416

WIPO PCT

[ST. 10/C]: [JP2004-071416]

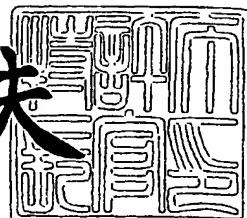
出願人
Applicant(s): シャープ株式会社

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2004年 5月27日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



【書類名】 特許願
【整理番号】 1040183
【提出日】 平成16年 3月12日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 15/00
【発明者】
 【住所又は居所】 大阪府大阪市阿倍野区長池町 22番22号 シャープ株式会社内
 【氏名】 平田 真章
【発明者】
 【住所又は居所】 大阪府大阪市阿倍野区長池町 22番22号 シャープ株式会社内
 【氏名】 松山 哲也
【発明者】
 【住所又は居所】 大阪府大阪市阿倍野区長池町 22番22号 シャープ株式会社内
 【氏名】 熊谷 圭司
【発明者】
 【住所又は居所】 大阪府大阪市阿倍野区長池町 22番22号 シャープ株式会社内
 【氏名】 仁田 壮一
【発明者】
 【住所又は居所】 大阪府大阪市阿倍野区長池町 22番22号 シャープ株式会社内
 【氏名】 音川 英之
【特許出願人】
 【識別番号】 000005049
 【住所又は居所】 大阪府大阪市阿倍野区長池町 22番22号
 【氏名又は名称】 シャープ株式会社
【代理人】
 【識別番号】 100064746
 【弁理士】
 【氏名又は名称】 深見 久郎
【選任した代理人】
 【識別番号】 100085132
 【弁理士】
 【氏名又は名称】 森田 俊雄
【選任した代理人】
 【識別番号】 100083703
 【弁理士】
 【氏名又は名称】 仲村 義平
【選任した代理人】
 【識別番号】 100096781
 【弁理士】
 【氏名又は名称】 堀井 豊
【選任した代理人】
 【識別番号】 100098316
 【弁理士】
 【氏名又は名称】 野田 久登
【選任した代理人】
 【識別番号】 100109162
 【弁理士】
 【氏名又は名称】 酒井 将行

【先の出願に基づく優先権主張】

【出願番号】 特願2003-150545

【出願日】 平成15年 5月28日

【先の出願に基づく優先権主張】

【出願番号】 特願2003-190785

【出願日】 平成15年 7月 3日

【先の出願に基づく優先権主張】

【出願番号】 特願2003-430127

【出願日】 平成15年12月25日

【手数料の表示】

【予納台帳番号】 008693

【納付金額】 21,000円

【提出物件の目録】

【物件名】 特許請求の範囲 1

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0208500

【書類名】特許請求の範囲**【請求項1】**

アプリケーションの記述を参照して、前記アプリケーションを実行する際に必要な機能手段を特定するための情報であるコネクタを取得するコネクタ取得手段と、

前記コネクタに記述される前記機能手段の存在場所へのアクセス情報に基づいて、前記存在場所へアクセスして前記コネクタで特定される機能手段を利用する機能利用手段とを備える、サービス利用端末。

【請求項2】

前記コネクタに記述されるアクセス情報は、前記存在場所へアクセスするためのURI (Uniform Resource Identifier) である、請求項1に記載のサービス利用端末。

【請求項3】

前記機能利用手段は、前記コネクタに定義される情報を前記コネクタに特定される機能手段に渡して前記機能手段から結果を得る、請求項1に記載のサービス利用端末。

【請求項4】

前記コネクタにはデータ変換情報が含まれ、

前記機能利用手段は、前記アプリケーションから得られるデータを前記データ変換情報に基づいて変換して前記機能手段に渡す、請求項3に記載のサービス利用端末。

【請求項5】

前記コネクタにはデータ変換情報が含まれ、

前記機能利用手段は、前記機能手段から得られる結果を前記データ変換情報に基づいて変換して前記アプリケーションに渡す、請求項3または4に記載のサービス利用端末。

【請求項6】

前記アプリケーションは、前記機能利用手段から渡された、前記機能利用手段によって変換された前記機能手段から得られる結果を出力するアプリケーションである、請求項5に記載のサービス利用端末。

【請求項7】

前記アプリケーションを実行する際に、前記アプリケーションに固有の識別情報と前記コネクタに固有の識別情報とを比較して、前記コネクタを用いて前記機能手段を利用するとの可否を判定する第1のコネクタ判定手段をさらに備える、請求項1～6のいずれかに記載のサービス利用端末。

【請求項8】

前記アプリケーションは、当該サービス利用端末に応じてカスタマイズされた固有情報を含み、

前記コネクタは、当該サービス利用端末に応じてカスタマイズされた固有情報を含み、

前記第1のコネクタ判定手段は、前記カスタマイズされた前記アプリケーションに固有の識別情報と前記コネクタに固有の識別情報とを比較して、当該サービス利用端末において前記アプリケーションを実行する際に前記コネクタを用いて前記機能手段を特定するとの可否を判定する、請求項7に記載のサービス利用端末。

【請求項9】

前記コネクタ取得手段は、他の装置にアクセスして前記他の装置に保持されるコネクタを取得し、

前記取得したコネクタを記憶するコネクタ保存手段をさらに備える、請求項1～8のいずれかに記載のサービス利用端末。

【請求項10】

前記アプリケーションには、前記アプリケーションを有効とする期間の情報と前記アプリケーションの利用可能回数との少なくとも一方を備える有効情報が含まれ、

前記有効情報を参照して前記アプリケーションの実行の可否を判定するアプリケーション判定手段をさらに備える、請求項1～9のいずれかに記載のサービス利用端末。

【請求項11】

前記コネクタには、前記コネクタを有効とする期間の情報と前記コネクタの利用可能回

数との少なくとも一方を備える有効情報が含まれ、

前記アプリケーションを実行する際に前記有効情報を参照して、前記コネクタを用いて前記機能手段を特定することの可否を判定する第2のコネクタ判定手段をさらに備える、請求項1～10のいずれかに記載のサービス利用端末。

【請求項12】

前記第2のコネクタ判定手段で前記コネクタを用いて前記機能手段を特定することが不可と判定された場合に、前記コネクタ取得手段で新たなコネクタを取得する、請求項11に記載のサービス利用端末。

【請求項13】

前記アプリケーションの記述は、前記コネクタを取得する取得先、および前記コネクタに固有の情報と前記コネクタで特定される前記機能手段に関する情報との少なくとも一方を備えるコネクタ条件を含み、

前記コネクタ取得手段は、前記取得先から、前記コネクタ条件を満たすコネクタを取得する、請求項1～12のいずれかに記載のサービス利用端末。

【請求項14】

前記コネクタ取得手段は、前記アプリケーションを実行する際に、前記コネクタ取得手段で取得するコネクタとして複数の異なるコネクタのうちから所定のコネクタを選択するコネクタ選択手段を含み、

前記機能利用手段は、前記複数の異なるコネクタで特定される複数の異なる機能手段のうちから、前記選択された所定のコネクタで特定される機能手段を利用する、請求項1～13のいずれかに記載のサービス利用端末。

【請求項15】

前記アプリケーションを取得するアプリケーション取得手段をさらに備える、請求項1～14のいずれかに記載のサービス利用端末。

【請求項16】

請求項1～15のいずれかに記載のサービス利用端末を含む、携帯電話端末。

【請求項17】

請求項1～15のいずれかに記載のサービス利用端末を含む、テレビジョン受像端末。

【請求項18】

アプリケーションを実行する際に必要な機能手段を特定するための、前記機能手段の存在場所へのアクセス情報を含む情報であるコネクタを保存するコネクタ保存手段と、

前記コネクタ保存手段に保存される前記コネクタを管理するコネクタ管理手段と、

サービス利用端末より前記コネクタの要求を受けて前記サービス利用端末に対して前記要求されたコネクタを送信するコネクタ送信手段とを備える、コネクタ提供サーバ。

【請求項19】

前記コネクタには、前記アプリケーションに対応した識別情報が含まれる、請求項18に記載のコネクタ提供サーバ。

【請求項20】

前記コネクタには、前記コネクタを有効とする期間の情報と前記コネクタの利用可能回数との少なくとも一方を備える有効情報が含まれる、請求項18または19に記載のコネクタ提供サーバ。

【請求項21】

前記コネクタに記述されるアクセス情報は、前記存在場所へアクセスするためのURI(Uniform Resource Identifier)である、請求項18～20のいずれかに記載のコネクタ提供サーバ。

【請求項22】

前記コネクタを要求した前記サービス利用端末の端末情報を取得する端末管理手段と、

前記端末情報に基づいて前記要求されたコネクタを前記サービス利用端末に対して送信するか否かを判定する判定手段とをさらに備える、請求項18～21のいずれかに記載のコネクタ提供サーバ。

【請求項23】

前記コネクタを要求した前記サービス利用端末の端末情報を取得する端末管理手段と、前記端末情報に基づいて前記要求されたコネクタを編集する第1の編集手段とをさらに備える、請求項18～22のいずれかに記載のコネクタ提供サーバ。

【請求項24】

前記機能手段の存在場所の形態に応じて、前記要求されたコネクタを編集する第2の編集手段をさらに備える、請求項18～23のいずれかに記載のコネクタ提供サーバ。

【請求項25】

前記第2の編集手段は、前記機能手段の存在場所が複数ある場合に、前記コネクタに記述されるアクセス情報を前記存在場所の内の所定の存在場所へのアクセス情報に書換える、請求項24に記載のコネクタ提供サーバ。

【請求項26】

前記第2の編集手段は、前記コネクタに記述されるアクセス情報を当該コネクタ提供サーバへのアクセス情報に書換え、

前記書換えられたアクセス情報に基づいた前記機能手段を必要とするサービス利用端末からの当該コネクタ提供サーバへのアクセスを、前記機能手段の存在場所へ中継する中継手段をさらに備える、請求項24に記載のコネクタ提供サーバ。

【請求項27】

サービス利用端末においてアプリケーションを実行する際に必要な機能手段を特定するための前記サービス利用端末の記憶手段に格納される情報であり、前記アプリケーションの記述から特定される情報であるコネクタデータのデータ構造であって、

前記サービス利用端末に、前記アプリケーションの記述を参照して前記コネクタデータを特定させるための前記コネクタデータの識別情報と、

前記サービス利用端末が前記機能手段の存在場所へアクセスするためのアクセス情報と、

前記サービス利用端末が前記アクセス情報にしたがって前記存在場所へアクセスして、前記機能手段に渡す情報を定義する入力定義情報と、

前記機能手段に渡した情報に対して、前記サービス利用端末が前記機能手段から受取る情報を定義する出力定義情報とを含む、コネクタデータのデータ構造。

【請求項28】

前記アクセス情報は、前記サービス利用端末が前記存在場所へアクセスするためのURI (Uniform Resource Identifier) である、請求項27に記載のコネクタデータのデータ構造。

【請求項29】

前記サービス利用端末における前記コネクタデータの利用を有効とする期間の情報と、前記サービス利用端末における前記コネクタの利用可能回数との少なくとも一方を備える有効情報をさらに含み、前記有効情報は前記サービス利用端末に対して、前記コネクタデータを用いて前記機能手段を特定することの可否の判定を可能とする、請求項27または28に記載のコネクタデータのデータ構造。

【請求項30】

アプリケーションを識別するアプリケーションの識別情報をさらに含み、前記アプリケーションの識別情報は、前記サービス利用端末に対して、前記アプリケーションを実行する際に前記コネクタデータを用いて前記機能手段を特定することの可否の判定を可能とする、請求項27～29のいずれかに記載のコネクタデータのデータ構造。

【請求項31】

サービス利用端末に応じてカスタマイズされた固有情報をさらに含み、前記固有情報は、前記サービス利用端末に対して、アプリケーションを実行する際に前記コネクタデータを用いて前記機能手段を特定することの可否の判定を可能とする、請求項27～30のいずれかに記載のコネクタデータのデータ構造。

【請求項32】

前記アクセス情報は、前記コネクタデータを提供するコネクタ提供サーバにおいて編集可能である、請求項27～31のいずれかに記載のコネクタデータのデータ構造。

【書類名】明細書

【発明の名称】サービス利用端末、携帯電話端末、テレビジョン受像端末、コネクタ提供サーバ、およびコネクタデータのデータ構造

【技術分野】

【0001】

この発明はサービス利用端末、携帯電話端末、テレビジョン受像端末、コネクタ提供サーバ、およびコネクタデータのデータ構造に関し、特に、いわゆるWebサービスといわれるネットワーク上で提供される機能を、利便性高く、かつ安全にユーザに提供できるサービス利用端末、携帯電話端末、テレビジョン受像端末、コネクタ提供サーバ、およびコネクタデータのデータ構造に関する。

【背景技術】

【0002】

携帯電話に代表されるような端末装置（以下、単に端末という）の、最近の高機能化により、多くの機能が端末内部に組込まれ、複雑な処理を端末だけで実行することが可能となった。しかし、端末には、依然、記憶容量・処理速度に限界があり、大量のデータを用いた複雑な処理を端末だけで行なうことは難しい。

【0003】

また、近年インターネットの普及に伴ない、パーソナルコンピュータや携帯電話を用いてネットワーク上の各種サービスを利用することが可能となった。このようなサービスには、たとえばウェブページを閲覧するといった単純なものから、ネットワーク上に存在する大量の記憶容量と高速な処理速度とを持ったサーバコンピュータにより提供される機能を利用するというものまで様々な形態が含まれる。特に、後者の、ネットワーク上に存在する機能を利用するという枠組みはWebサービスと呼ばれ、目覚しい速度で普及が進んでいる。

【0004】

Webサービスについてはたとえば非特許文献1などに記述されており、図39に示されるように、パーソナルコンピュータ（PC）等の端末とサーバ（Web Server）との間でRPC（Remote Procedure Call）を行なうためのデータの転送方法を定義したプロトコルであるSOAP（Simple Object Access Protocol）、サーバの提供する機能の入出力を定義するWSDL（Web Service Description Language）、およびネットワーク上に存在するサーバの機能を検索するために定義されたUDDI（Universal Description, Discovery and Integration）という国際標準技術に基づいて、各種サーバおよび端末間に偏在する機能を利用する枠組みに対する呼称である。

【0005】

Webサービスに代表される、ネットワークを用いたサービスの提供および利用方法として、特許文献1においてネットワークサービス提供システムが開示されている。特許文献1においては、携帯電話を情報のリモートコントローラとして使用することにより、ネットワーク上のあらゆるサービスを携帯電話から利用可能とするネットワークサービス提供システムを開示している。

【特許文献1】特開2003-101589号公報

【非特許文献1】本俊也、「詳細Webサービス構築」、初版、ソフトバンクパブリッシング株式会社、2003年4月、p. 2-16

【発明の開示】

【発明が解決しようとする課題】

【0006】

しかしながら、上述の特許文献1に開示されるシステムでは、ネットワーク上に存在する端末外の機能は一意に扱えるが、端末内部に存在する機能までは同じ形式で扱うことができないという問題がある。

【0007】

また、非特許文献1に記述されているような一般的なWebサービスにおいては、端末

内の特定のアプリケーションから利用可能である端末外部の機能および端末内部の機能を設定できないため、たとえばA社の提供する端末外部の機能はA社の提供するアプリケーションからのみ利用できる、といった制限を設けることができないという問題もある。

【0008】

さらに、端末からネットワークに存在するすべてのサービスを利用できてしまうため、セキュリティ的に問題のあるサービスや機能にも接続して利用することが可能であるという問題がある。たとえば、端末内部のデータを破壊してしまうようなサービスにはアクセスさせない、すなわち安全であると承認されたサービスや機能のみを端末から利用可能にすることができないという問題がある。

【0009】

さらに、従来のアプリケーションにおいては、アプリケーションのユーザインタフェースと端末外部のサービスや端末内部の機能に接続する機能とが分離されていないため、同じインタフェースを用いて複数の異なるサービスを使い分けて利用することができないという問題があった。

【0010】

本発明は、このような問題を解決するためになされたものであって、利便性が高く、安全性の高いWebサービスをユーザに提供できるサービス利用端末、携帯電話端末、テレビジョン受像端末、コネクタ提供サーバ、およびコネクタデータのデータ構造を提供することを目的とする。

【課題を解決するための手段】

【0011】

上記目的を達成するために、本発明のある局面に従うと、サービス利用端末は、アプリケーションの記述を参照して、アプリケーションを実行する際に必要な機能手段を特定するための情報であるコネクタを取得するコネクタ取得手段と、コネクタに記述される機能手段の存在場所へのアクセス情報に基づいて、存在場所へアクセスしてコネクタで特定される機能手段を利用する機能利用手段とを備える。

【0012】

さらに、コネクタに記述されるアクセス情報は、存在場所へアクセスするためのURI(Uniform Resource Identifier)であることが好ましい。

【0013】

また、機能利用手段は、コネクタに定義される情報をコネクタに特定される機能手段に渡して、その機能手段から結果を得ることが好ましい。

【0014】

さらに、コネクタにはデータ変換情報が含まれ、機能利用手段は、アプリケーションから得られるデータをデータ変換情報に基づいて変換して機能手段に渡すことが好ましい。

【0015】

さらに、コネクタにはデータ変換情報が含まれ、機能利用手段は、機能手段から得られる結果をデータ変換情報に基づいて変換してアプリケーションに渡すことが好ましい。

【0016】

さらに、そのアプリケーションは、機能利用手段から渡された、機能利用手段によって変換された機能手段から得られる結果を出力するアプリケーションであることがより好ましい。

【0017】

また、サービス利用端末は、アプリケーションを実行する際に、アプリケーションに固有の識別情報とコネクタに固有の識別情報とを比較して、コネクタを用いて機能手段を利用するとの可否を判定する第1のコネクタ判定手段をさらに備えることが好ましい。

【0018】

さらに、アプリケーションは、当該サービス利用端末に応じてカスタマイズされた固有情報を含み、コネクタは、当該サービス利用端末に応じてカスタマイズされた固有情報を含み、第1のコネクタ判定手段は、カスタマイズされたアプリケーションに固有の識別情

報とコネクタに固有の識別情報を比較して、当該サービス利用端末においてアプリケーションを実行する際にコネクタを用いて機能手段を特定することの可否を判定することが好ましい。

【0019】

また、上述のコネクタ取得手段は、他の装置にアクセスして他の装置に保持されるコネクタを取得し、サービス利用端末は、取得したコネクタを記憶するコネクタ保存手段をさらに備えることが好ましい。

【0020】

また、アプリケーションには、アプリケーションを有効とする期間の情報とアプリケーションの利用可能回数との少なくとも一方を備える有効情報が含まれ、サービス利用端末は、有効情報を参照してアプリケーションの実行の可否を判定するアプリケーション判定手段をさらに備えることが好ましい。

【0021】

また、コネクタには、コネクタを有効とする期間の情報とコネクタの利用可能回数との少なくとも一方を備える有効情報が含まれ、サービス利用端末は、アプリケーションを実行する際に有効情報を参照して、コネクタを用いて機能手段を特定することの可否を判定する第2のコネクタ判定手段をさらに備えることが好ましい。

【0022】

さらに、サービス利用端末は、第2のコネクタ判定手段でコネクタを用いて機能手段を特定することが不可と判定された場合に、コネクタ取得手段で新たなコネクタを取得することがより好ましい。

【0023】

また、アプリケーションの記述は、コネクタを取得する取得先、およびコネクタに固有の情報とコネクタで特定される機能手段に関する情報との少なくとも一方を備えるコネクタ条件を含み、コネクタ取得手段は、取得先から、コネクタ条件を満たすコネクタを取得することが好ましい。

【0024】

また、上述のコネクタ取得手段は、アプリケーションを実行する際に、コネクタ取得手段で取得するコネクタとして複数の異なるコネクタのうちから所定のコネクタを選択するコネクタ選択手段を含み、機能利用手段は、複数の異なるコネクタで特定される複数の異なる機能手段のうちから、選択された所定のコネクタで特定される機能手段を利用するこことが好ましい。

【0025】

また、サービス利用端末は、アプリケーションを取得するアプリケーション取得手段をさらに備えることが好ましい。

【0026】

本発明の他の局面に従うと、携帯電話端末は、上述のいずれかのサービス利用端末を含む。

【0027】

本発明のさらに他の局面に従うと、テレビジョン受像端末は、上述のいずれかのサービス利用端末を含む。

【0028】

本発明のさらに他の局面に従うと、コネクタ提供サーバは、アプリケーションを実行する際に必要な機能手段を特定するための、機能手段の存在場所へのアクセス情報を含む情報であるコネクタを保存するコネクタ保存手段と、コネクタ保存手段に保存されるコネクタを管理するコネクタ管理手段と、サービス利用端末よりコネクタの要求を受けてサービス利用端末に対して要求されたコネクタを送信するコネクタ送信手段とを備える。

【0029】

また、上述のコネクタには、アプリケーションに対応した識別情報が含まれることが好ましい。

【0030】

また、上述のコネクタには、コネクタを有効とする期間の情報とコネクタの利用可能回数との少なくとも一方を備える有効情報が含まれることが好ましい。

【0031】

また、上述のコネクタに記述されるアクセス情報は、存在場所へアクセスするためのURI (Uniform Resource Identifier) であることが好ましい。

【0032】

また、コネクタ提供サーバは、コネクタを要求したサービス利用端末の端末情報を取得する端末管理手段と、端末情報に基づいて要求されたコネクタをサービス利用端末に対して送信するか否かを判定する判定手段とをさらに備えることが好ましい。

【0033】

また、コネクタ提供サーバは、コネクタを要求したサービス利用端末の端末情報を取得する端末管理手段と、端末情報に基づいて要求されたコネクタを編集する第1の編集手段とをさらに備えることが好ましい。

【0034】

また、コネクタ提供サーバは、機能手段の存在場所の形態に応じて、要求されたコネクタを編集する第2の編集手段をさらに備えることが好ましい。

【0035】

さらに第2の編集手段は、機能手段の存在場所が複数ある場合に、コネクタに記述されるアクセス情報を、存在場所の内の、所定の存在場所へのアクセス情報に書換えることが好ましい。

【0036】

あるいは、第2の編集手段は、コネクタに記述されるアクセス情報を当該コネクタ提供サーバへのアクセス情報に書換え、コネクタ提供サーバは、書換えられたアクセス情報に基づいた機能手段を必要とするサービス利用端末からの当該コネクタ提供サーバへのアクセスを、機能手段の存在場所へ中継する中継手段をさらに備えることが好ましい。

【0037】

本発明のさらに他の局面に従うと、コネクタデータのデータ構造は、サービス利用端末においてアプリケーションを実行する際に必要な機能手段を特定するためのサービス利用端末の記憶手段に格納される情報であり、アプリケーションの記述から特定される情報であるコネクタデータのデータ構造であって、サービス利用端末に、アプリケーションの記述を参照してコネクタデータを特定させるためのコネクタデータの識別情報と、サービス利用端末が機能手段の存在場所へアクセスするためのアクセス情報と、サービス利用端末がアクセス情報にしたがって存在場所へアクセスして、機能手段に渡す情報を定義する入力定義情報と、機能手段に渡した情報に対して、サービス利用端末が機能手段から受取る情報を定義する出力定義情報とを含む。

【0038】

また、上述のアクセス情報は、サービス利用端末が存在場所へアクセスするためのURI (Uniform Resource Identifier) であることが好ましい。

【0039】

また、コネクタデータのデータ構造は、サービス利用端末におけるコネクタデータの利用を有効とする期間の情報と、サービス利用端末におけるコネクタの利用可能回数との少なくとも一方を備える有効情報をさらに含み、有効情報はサービス利用端末に対して、コネクタデータを用いて機能手段を特定することの可否の判定を可能とすることが好ましい。

【0040】

また、コネクタデータのデータ構造は、アプリケーションを識別するアプリケーションの識別情報をさらに含み、アプリケーションの識別情報は、サービス利用端末に対して、アプリケーションを実行する際にコネクタデータを用いて機能手段を特定することの可否の判定を可能とすることが好ましい。

【0041】

また、コネクタデータのデータ構造は、サービス利用端末に応じてカスタマイズされた固有情報をさらに含み、固有情報は、サービス利用端末に対して、アプリケーションを実行する際にコネクタデータを用いて機能手段を特定することの可否の判定を可能とすることが好ましい。

【0042】

また、上述のアクセス情報は、コネクタデータを提供するコネクタ提供サーバにおいて編集可能であることが好ましい。

【発明を実施するための最良の形態】

【0043】

以下に、図面を参照しつつ、本発明の実施の形態について説明する。以下の説明では、同一の部品および構成要素には同一の符号を付してある。それらの名称および機能も同じである。したがってそれらについての詳細な説明は繰返さない。

【0044】

図1は、本実施の形態におけるサービス提供システムの構成の具体例を示す図である。本実施の形態におけるサービス提供システムは、本発明にかかるサービス利用端末と、サービス利用端末に対して機能を提供するサービス提供サーバとを含んで構成される。

【0045】

図1を参照して、本実施の形態におけるサービス利用端末であるサービス利用パーソナルコンピュータ（以下、PCという）100aは、ネットワーク200を介してサービス利用端末に対して機能を提供するサービス提供サーバ210に接続される。

【0046】

また、ネットワーク200には、携帯電話網230、ホームネットワーク260、およびコネクタ提供サーバ280がさらに接続される。そして、本実施の形態におけるサービス利用端末であるサービス利用テレビ100bは、いわゆるホームネットワーク260、およびネットワーク200を介してサービス提供サーバ210に接続される。

【0047】

さらに、ネットワーク200に接続される携帯電話網230には携帯電話基地局240が接続される。そして、本実施の形態におけるサービス利用端末であるサービス利用携帯電話100cは、携帯電話基地局240、携帯電話網230、およびネットワーク200を介してサービス提供サーバ210に接続される。

【0048】

なお、本実施の形態において、サービス利用端末は、ネットワーク200を介してサービス提供サーバ210にアクセスしてサービス提供サーバ210が提供する機能を利用するサービス利用PC100a、サービス利用テレビ100b、およびサービス利用携帯電話100cであるものとし、これらの装置を総称してサービス利用端末100とするが、本発明のサービス利用端末はこれらの装置に限定されず、ネットワークを介してサービス提供サーバ210の提供する機能を利用することのできるその他の端末であってもよい。

【0049】

たとえば、PDA (Personal Digital Assistants: 携帯情報端末) や、ビデオレコーダ、DVD (Digital Video Disc) レコーダ、ハードディスクビデオレコーダなどのPVR (Personal Video Recorder) 、デジタル放送チューナ、ホームサーバなどの情報家電機器や、冷蔵庫、洗濯機、電子レンジなどのいわゆる白物家電機器などであっても、ネットワークサービスを利用可能な端末であれば、すべて、本発明のサービス利用端末とみなすことが可能である。

【0050】

サービス提供サーバ210は、サービス利用端末100から利用できる機能を提供するサーバである。サービス提供サーバ210がサービス利用端末100に対して提供するサービスは、アプリケーションを含む機能であって、具体的には、プログラム等のアプリケーションや、サービス利用端末100においてアプリケーションを実行するために必要と

なる機能（文字表示機能、文字装飾機能、画像取得機能、アニメーション作成機能等）が該当する。

【0051】

また、コネクタ提供サーバ280は、サービス利用端末100からサービス提供サーバ210の提供する機能を利用するため必要となるコネクタをサービス利用端末100へ提供するサーバである。コネクタ提供サーバ280がサービス利用端末100に対して提供するコネクタとは、サービス利用端末100においてアプリケーションを実行する際に必要となる機能を特定するための情報である。なお、コネクタについては、後に具体例を挙げて説明する。

【0052】

次に、図2に、本実施の形態におけるサービス利用端末100のハードウェア構成の具体例を示す。図2を参照して、本実施の形態におけるサービス利用端末100は、CPU (Central Processing Unit) 等から構成されて、このサービス利用端末100全体を制御する制御部12と、他の装置と通信するための通信部11と、ROM (Read Only Memory) やRAM (Random Access Memory) などから構成されて、制御部12において実行されるプログラムや、そのプログラムの中間データおよび他のコンピュータから受信したデータ等を記憶する記憶部13とを含んで構成される。

【0053】

さらに上述の入出力部14は、「1」、「2」などの数字ボタンおよび「R」、「L」などの方向ボタンなどを含む操作部141と、ユーザに情報を表示するLCD (Liquid Crystal Display) などの表示部142と、音声を入力するマイクロフォンなどの音声入力部143と、音声を出力するスピーカなどの音声出力部144とを含んで構成される。

【0054】

また、上述の記憶部13は、一般のユーザの操作により記憶内容を変更可能な領域である一般ユーザ変更可能領域131と、一般のユーザの操作によってその記憶内容を変更することができない領域である一般ユーザ変更不可領域132とを含んで構成される。一般ユーザ変更不可領域132には、当該サービス利用端末100が携帯電話である場合には電話番号などの、当該サービス利用端末100固有の識別子や、後述するアプリケーションやコネクタを取得するためのサーバのアドレスといった、一般のユーザにより変更されなければならないデータが主に記憶される。ただし、一般ユーザ変更不可領域132に記憶されたデータを変更することを認められたサービスの提供者や端末のメンテナンス実行者およびプログラム等により、一般ユーザ変更不可領域132に対するデータの追加・削除、およびデータの変更が行なわれるための手段が存在してもよい。

【0055】

なお、図2に示されるハードウェア構成は、一般的な携帯電話のハードウェア構成であって、本発明にかかるサービス利用端末100のハードウェア構成は、図2に示されるハードウェア構成に限定されるものではない。

【0056】

次に、図3に、本実施の形態におけるサービス利用端末100の機能構成の具体例を示す。図3を参照して、本実施の形態におけるサービス利用端末100は、アプリケーション管理部110、コネクタ管理部120、外部機能管理部130、内部機能管理部140、通信制御部150、および記憶領域160を含んで構成される。これらの機能は、制御部12が、記憶部13に記憶されるプログラムを読み出し、そのプログラムを実行して図2に示されるハードウェア構成の各部を制御することによって発揮される。

【0057】

記憶領域160は、上述の記憶部13の一般ユーザ変更不可領域132を含んで構成される領域であって、上述の如く、サービス提供サーバ210へのアクセス情報（アドレス）や、コネクタ提供サーバ280へのアクセス情報（アドレス）や、当該サービス利用端末100の識別情報などが記憶されている。

【0058】

アプリケーション管理部110は、サービス利用端末100においてアプリケーションを実行する際に、アプリケーションデータを解析して必要とするコネクタを決定する。そして、コネクタ管理部120に対して、コネクタを要求する。なお、アプリケーション管理部110については、後にさらに詳細に説明する。

【0059】

コネクタ管理部120は、アプリケーション管理部110から要求されたコネクタを取得する。その際、要求されたコネクタが当該サービス利用端末100に含まれない場合には、記憶領域160に記憶されているコネクタ提供サーバ280へのアクセス情報を読み出し、そのアクセス情報にしたがってコネクタ提供サーバ280へアクセスして要求されたコネクタを取得する。なお、コネクタ提供サーバ280へのアクセス情報はアプリケーション内の記述に含まれてもよい。通信制御部150は、その際の、サービス利用端末100とコネクタ提供サーバ280との通信を制御する。

【0060】

そして、コネクタ管理部120は、取得したコネクタデータを解析して、当該アプリケーションを実行する際に必要とする機能とその存在場所とを決定する。必要とする機能がサービス提供サーバ210に存在する機能である場合、すなわち、サービス提供サーバ210が提供する機能である場合には、コネクタ管理部120は、外部機能管理部130に対してその機能を要求する。また、必要とする機能が当該サービス利用端末100に含まれる機能である場合には、コネクタ管理部120は、内部機能管理部140に対してその機能を要求する。なお、コネクタ管理部120については、後にさらに詳細に説明する。

【0061】

外部機能管理部130は、記憶領域160に記憶されているサービス提供サーバ210へのアクセス情報を読み出し、そのアクセス情報にしたがってサービス提供サーバ210へアクセスする。そして、コネクタ管理部120から要求された機能を利用する。通信制御部150は、その際の、サービス利用端末100とサービス提供サーバ210との通信を制御する。

【0062】

内部機能管理部140は、当該サービス利用端末100に含まれる機能であって、コネクタ管理部120から要求された機能を利用する。

【0063】

次に、図4に、アプリケーション管理部110の構成の具体例を示す。図4を参照して、アプリケーション管理部110は、アプリケーションデータ保存領域300とアプリケーションデータ管理部320とを含んで構成される。

【0064】

アプリケーションデータ保存領域300は、アプリケーションデータ310を保存する。ここでアプリケーションデータ310は、コンピュータにより実行可能なプログラムであってもよいし、スクリプト解釈装置により解釈され実行されるスクリプトファイルであってもよい。

【0065】

アプリケーションデータ管理部320は、アプリケーションデータ310の追加、削除、およびネットワークからのダウンロードなどのアプリケーションデータ保存領域300と、アプリケーションの実行、および停止などのアプリケーションデータ310の動作とを管理する。

【0066】

さらに、図5に、アプリケーションデータ310の具体例を示す。図5を参照して、アプリケーションデータ310は、アプリケーション名401、当該アプリケーションに固有の情報であるアプリケーション識別子402、当該アプリケーションを実行可能な期限を示す有効期限403、当該アプリケーションを実行可能な回数を示す最大利用回数404、およびアプリケーション405を含んで構成される。アプリケーション405は、実行可能なプログラムであってもよいし、スクリプト解釈装置により解釈され実行されるス

クリプトファイルであってもよい。

【0067】

図6は、図5に示されたアプリケーションデータ310のうち、アプリケーション名401、アプリケーション識別子402、有効期限403、および最大利用回数404の具体例を示す図である。図6に示される具体例においては、アプリケーション名401は「Multimedia Message」と設定されている。また、アプリケーション識別子402は「提供元が“AAA Coop.”、一意に割当てられた番号が“HG3490S2345”」と設定されている。また、有効期限403は「2004年4月1日まで有効」と設定されている。また、最大利用回数404は「100回有効」と設定されている。

【0068】

さらに、アプリケーション405がスクリプトにより記述されている場合のアプリケーション405の具体例を図7に示す。図7に示される具体例は、アプリケーション405が拡張されたSMIL (Synchronized Multimedia Integration Language) により記述されている場合の具体例であるが、アプリケーション405は他の書式によって記述されていても構わない。

【0069】

図7を参照して、記述601はアプリケーションの画面表示を設定する記述であって、具体的には、「幅150、高さ100で、左から25、上から10の位置に表示する」Image領域についての表示設定と、同様にText領域についての表示設定と、「幅40、高さ30で左から10、上から160の位置に表示する」TextArtボタンについての表示設定と、同様にCameraボタンについての表示設定と、同様にFileボタンについての表示設定と、同様にSendボタンについての表示設定とを行なっている。

【0070】

また、記述602はアクションの動作を設定する記述であって、具体的には、「検索範囲を“Terminal (サービス利用端末100内部)”として検索されたTextArtという名前のコネクタを利用し、該当する機能にはText領域の値を渡し、その機能から受取った結果をText領域に表示する」TextArtというアクションの動作と、「検索範囲を“Terminal”として検索されたCameraという名前のコネクタを利用し、該当する機能には値を渡す必要はなく、その機能から受取った結果はImage領域に表示する」Cameraというアクションの動作と、「検索範囲を“Terminal”として検索されたFileという名前のコネクタを利用し、該当する機能には値を渡す必要はなく、その機能から受取った結果はImage領域に表示する」Fileというアクションの動作と、「検索範囲を“Terminal”として検索されたSendという名前のコネクタを利用し、該当する機能には値を渡す必要はなく、その機能から受取った結果は画面表示を更新しない」Sendというアクションの動作とを設定している。なお、図7には図示されていないが、コネクタの検索範囲をコネクタ提供サーバ280とするときや、コネクタの検索範囲をサービス利用端末100内部およびコネクタ提供サーバ280 (“Server”) とするときには、記述602にその旨を示す記述が含まれる。

【0071】

また、記述603はボタンが押された時の動作を設定する記述であって、具体的には、「TextArtというアクションが実行される」というTextArtボタンが押された場合の動作と、「Cameraというアクションが実行される」というCameraボタンが押された場合の動作と、「Fileというアクションが実行される」というFileボタンが押された場合の動作と、「Sendというアクションが実行される」というSendボタンが押された場合の動作とを設定している。

【0072】

なお、図7に示されるアプリケーション405がサービス利用端末100において実行

された際の画面表示については、後に具体例を挙げて説明する。

【0073】

次に、図8に、図3に示されるサービス利用端末100のコネクタ管理部120の構成の具体例を示す。図8を参照して、コネクタ管理部120は、コネクタデータ保存領域700とコネクタデータ管理部720とを含んで構成される。

【0074】

コネクタデータ保存領域700は、コネクタデータ710を保存する。

【0075】

コネクタデータ管理部720は、コネクタデータ710の追加、削除、およびネットワークからのダウンロードなどのコネクタデータ保存領域700の管理を行なう。また、コネクタデータ710の記述内容にしたがって、外部機能管理部130および内部機能管理部140の機能を利用する。

【0076】

さらに、図9に、コネクタデータ710の具体例を示す。図9を参照して、コネクタデータ710は、当該コネクタ自身の情報であるコネクタ名801、当該コネクタに固有の情報であるコネクタ識別子802、当該コネクタを利用可能な期限を示す有効期限803、および当該コネクタを利用可能な回数を示す最大利用回数804と、当該コネクタが決定する機能に対するデータの受け渡しを定義する入出力定義805とを含んで構成される。

【0077】

図10は、図9に示されたコネクタデータ710のうち、コネクタ名801、コネクタ識別子802、有効期限803、および最大利用回数804の具体例を示す図である。図10に示される具体例においては、コネクタ名801は「TextArt」と設定されている。また、コネクタ識別子802は「提供元が“AAA Coop.”、このコネクタを使えるアプリケーションの提供元が“AAA Coop.”、一意に割当てられた番号が“LS4369P3550”」と設定されている。また、有効期限803は「2004年4月1日まで有効」と設定されている。また、最大利用回数804は「100回有効」と設定されている。

【0078】

図10に示されるコネクタデータ710は、そのコネクタ識別子802において当該コネクタを利用可能なアプリケーションの提供元を設定することで、当該コネクタを利用可能なアプリケーションを制限していることを特徴としている。

【0079】

同様に、図11に、コネクタデータ710のうち、コネクタ名801、コネクタ識別子802、有効期限803、および最大利用回数804の他の具体例を示す。図11を参照して、コネクタ名801は「Camera」と設定されている。また、コネクタ識別子802は「提供元が“AAA Coop.”、このコネクタを使えるアプリケーションの提供元は指定なし（どのアプリケーションからも利用可能）、一意に割当てられた番号が“RE6589P8112”」と設定されている。また、有効期限803は「指定なし（常に利用可能）」と設定されている。また、最大利用回数804は「指定なし（何度でも利用可能）」と設定されている。

【0080】

さらに、コネクタデータ710に含まれる入出力定義805の具体例を図12に示す。図12に具体例が示される入出力定義805aは、サービス利用端末100においてアプリケーションを実行する際に、当該端末100以外の他の装置に含まれる機能、すなわち、サービス提供サーバ210の提供する機能を利用するための、当該機能に対するデータの受け渡しを定義するものである。すなわち、図12を参照して、記述2501は当該機能で利用するデータの型を定義しており、具体的には、“TextArtリクエスト”という名前で型が文字列であるデータと、“TextArtレスポンス”という名前で型がJpeg画像であるデータとを定義している。

【0081】

また、記述2502は当該機能について、機能の名前と当該機能が受取る入力データおよび当該機能が返す出力データの型とを定義しており、具体的には、機能の名前を“TextArt”、入力値として“TextArtリクエスト”という名前の型のデータを受取り、出力値として“TextArtレスポンス”という名前の型のデータを返すと定義している。

【0082】

また、記述2503は、当該機能が提供されている際の名前、当該機能の存在場所へのアクセス情報、およびそこで利用可能な機能を定義しており、具体的には、サービスの名前を“TextArtサービス”、当該機能の存在場所であるサービス提供サーバ210のアドレスをhttp://example.url/TextArt、および利用できる機能を“TextArt”と定義している。

【0083】

入出力定義805aが上述の記述2501～2503を含むことで、サービス利用端末100のコネクタデータ管理部720は、http://example.url/TextArtというアドレスでサービス提供サーバ210にアクセスし、サービス提供サーバ210に含まれるTextArtという機能に対して、TextArtリクエスト型のデータを渡すことにより、サービス提供サーバ210からTextArtレスポンス型のデータを得られることを認識できる。

【0084】

なお、入出力定義805aは上述の記述を含むものに限定されず、一般的によく用いられるWebサービスの入出力定義のための書式WSDL (Web Service Description Language) を用いた記述を含むものであってもよい。また、アクセス情報は、URLで示される形態に限定されず、サービス提供サーバ210へのアクセスに関する情報であるURI (Uniform Resource Identifier) であれば、他の形態であってもよい。

【0085】

さらに図13に、入出力定義805の他の具体例を示す。図13に具体例が示される入出力定義805bは、サービス利用端末100においてアプリケーションを実行する際に、当該端末に含まれる機能を利用するための、当該機能に対するデータの受け渡しを定義するものである。すなわち、図13を参照して、記述2601は当該機能で利用するデータの型を定義しており、具体的には“Cameraレスポンス”という名前で型がJpeg画像であるデータを定義している。

【0086】

また、記述2602は当該機能について、機能の名前と機能が受取る入力データおよび機能が返す出力データの型とを定義しており、具体的には、機能の名前を“Camera”、入力値を“Null”と設定しており、これは入力値を受取らないことを意味し、出力値として“Cameraレスポンス”という名前の型のデータを返すと定義している。

【0087】

図13に示される入出力定義805bは、図12に示された記述2503を含まない。すなわち、図13に示される入出力定義805bが、上述の記述2601, 2602を含むことで、サービス利用端末100のコネクタデータ管理部720は、当該サービス利用端末100に含まれるCameraという機能から、Cameraレスポンス型のデータを得られることを認識できる。言うまでもなく、当該サービス利用端末100に含まれる機能を利用するための入出力定義805bも、図12に示される入出力定義805aと同様に、機能が提供されている際の名前、機能の存在場所へのアクセス情報、およびそこで利用可能な機能を定義する記述2503を含んでもよい。その場合、記述2503に含まれる機能の存在場所へのアクセス情報として、サービス利用端末100自身へのアクセスを示す“Local”などの記載があることが好ましい。

【0088】

なお、サービス利用端末100に含まれる機能に対するデータの受け渡しを定義する入

出力定義805bもまた上述の記述を含むものに限定されず、一般的によく用いられるWebサービスの入出力定義のための書式WSDLを用いた記述を含むものであってもよい。

【0089】

さらに、コネクタデータ710は図9に示されるデータ構成に限定されず、たとえば、図43に示されるように、入出力定義805には複数の入出力定義1, 2が含まれていてもよい。これは、コネクタが、同様の働きを行なう機能に対して異なる個数の引数をサポートしている場合に該当する。たとえば、画像の合成処理を実行する機能を特定するコネクタで、2つの画像入力も、3つの画像入力も、または、3つ以上の画像入力もサポートしている場合がある。図43に示されるように入出力定義805が複数定義されている場合、1つのコネクタでこのような複数の入力パターンに対応することができる。

【0090】

さらに図43に示されるように、コネクタデータ710は図9に示されるデータ構成に加えて、最大待ち時間806や、転送許可数807や、処理負荷係数808や、利用金額809などを含んでもよい。

【0091】

コネクタデータ710が図43に示されるように最大待ち時間806を含むことで、サービス利用端末100がコネクタデータ710に特定されるサービス利用端末100の機能を利用する場合、またはネットワーク上のサービス提供データ210にアクセスして機能をダウンロードする場合、サービス利用端末100は、コネクタデータ710に含まれる最大待ち時間806を参考にして前記機能を利用する際の例外的な処理を行なうことができる。なお、コネクタデータ710に含まれる最大待ち時間806はあくまで参考の値であり、実装はサービス利用端末100、またはアプリケーション405に依存する。

【0092】

また、コネクタの中には、サービス利用端末間で一定の回数だけ転送を許可されたものも存在する。転送許可数807は、許可された転送回数を規定している。コネクタデータ710が図43に示されるように転送許可数807を含むことで、サービス利用端末100は所持しているコネクタデータ710が、転送可能なのか、不可能なのかを判断することができる。

【0093】

また、サービス利用端末100がコネクタデータ710に特定されるサービス利用端末100の機能を利用する場合、またはネットワーク上のサービス提供データ210にアクセスして機能をダウンロードする場合、サービス利用端末100が各々同様の機能を特定する複数のコネクタデータを所持しているとき、そのアプリケーションを実行する際にいずれの機能を利用してもよい場合がある。このようなとき、コネクタデータ710が図43に示されるように処理負荷係数808を含むことで、サービスの処理時間や自端末に必要となる負荷の指標となるこの処理負荷係数808を参考にして最適なコネクタを選択することができる。

【0094】

また、サービスの利用に応じて課金が発生する種類のコネクタも存在する。コネクタデータ710が図43に示されるように利用金額809を含むことで、サービス利用端末100において課金情報を把握することができる。サービス利用端末100の利用者は、搭載されているコネクタの課金情報を閲覧することができ、不要となった有料コネクタを破棄する判断に利用することができる。

【0095】

次に、図14に、図3に示されるサービス利用端末100の外部機能管理部130の構成の具体例を示す。図14を参照して、外部機能管理部130は、外部機能呼出部1101と外部機能結果解析部1102とを含んで構成される。

【0096】

外部機能呼出部1101は、サービス利用端末100の外部の他の装置、すなわちサー

ビス提供サーバ210に存在する機能（以下、外部機能という）をRPCにより呼出す。RPCによる外部機能の呼出しには、SOAPなどの一般的なプロトコルを用いることができる。

【0097】

外部機能結果解析部1102は、外部機能より得られた結果を解析する。外部機能より返却される結果は、前述のSOAP等、一般的なプロトコルにしたがった形式で受取ることができる。

【0098】

次に、図15に、図3に示されるサービス利用端末100の内部機能管理部140の構成の具体例を示す。図15を参照して、内部機能管理部140は、内部機能呼出部1201と内部機能結果解析部1202とを含んで構成される。

【0099】

内部機能呼出部1201は、当該サービス利用端末100内に保持される機能（プログラム、データへのアクセス）を呼出す。また、内部機能結果解析部1202は、当該サービス利用端末100内に保持される機能を実行した結果を受取り、その結果を解析する。

【0100】

次に、図40に、本実施の形態におけるコネクタ提供サーバ280のハードウェア構成の具体例を示す。図40を参照して、本実施の形態におけるコネクタ提供サーバ280は、CPU等のデバイスから構成され、このコネクタ提供サーバ280全体を制御する制御部22と、他の装置と通信するための通信部21と、ROMやRAMなどのメモリデバイスや、HDD等のディスクドライブなどから構成され、プログラムや、プログラムの実行に必要となる各種データの保存を行なう記憶部23と、外部から当該端末に対してデータを入力するためのA～Zや各種数字、記号のキーを備えたキーボードから構成される操作部24と、各種出力結果や操作内容の表示を行なうために、ディスプレイ装置を活用した表示部25とを含んで構成される。

【0101】

さらに上述の記憶部23は、主にプログラム等の固定データを保存するためのプログラム記憶領域231と、プログラムの中間データや他のコンピュータとのやり取りに必要となるデータの保存を行なうデータ記憶領域232とを含んで構成される。

【0102】

なお、図40に示されるハードウェア構成は、一般的なパソコンやワークステーション等のハードウェア構成であって、本発明にかかるコネクタ提供サーバ280のハードウェア構成は、図40に示されるハードウェア構成に限定されるものではない。たとえば、本実施の形態におけるコネクタ提供サーバ280の運用時には、操作部24と表示部25とが装備されていなくてもよい。

【0103】

次に、図41に、本実施の形態におけるコネクタ提供サーバ280の機能構成の具体例を示す。図41を参照して、本実施の形態におけるコネクタ提供サーバ280は、端末管理部510、コネクタ管理部520、通信制御部530、および記憶領域540を含んで構成される。これらの機能は、制御部22が、記憶部23のプログラム領域231に記憶されるプログラムを読み出し、読み出したプログラムに従って図40に記載の通信部21、記憶部23、および記憶部23のデータ領域232を制御することによって発揮される。

【0104】

記憶領域540は各種データを記憶部している場所であり、図40に記載の記憶部23にて構成される。

【0105】

端末管理部510は、サービス利用端末100からコネクタ提供サーバ280に対してコネクタ提供の要求があった際に、要求があったサービス利用端末100の端末情報の管理を行なう。具体的には、現在、接続中のサービス利用端末100を一意に識別するための識別情報等を、記憶部23のデータ領域231から構成される記憶領域540に格納し

ておき、前記識別情報に基づいて、通信制御部530に依頼を行ない、任意のサービス利用端末とのやり取りを実現する。

【0106】

コネクタ管理部520は、コネクタ提供サーバ280が提供可能なコネクタの管理を行なう。サービス利用端末100からの要求に応じて要求されているコネクタの判別、確認、および抽出を行なった後、通信部21と制御部22とを含んで構成される通信制御部530に送信すべきコネクタを送る。

【0107】

図42を用いて、コネクタ提供サーバ280のコネクタ管理部520についてさらに具体例を示す。図42を参照して、コネクタ管理部520は、コネクタデータ保存領域900とコネクタデータ管理部920とを含んで構成される。

【0108】

コネクタデータ保存領域900は、サービス利用端末100に提供するべきコネクタの実体であるコネクタデータ910を保存する。

【0109】

コネクタデータ管理部920は、コネクタデータ910の追加、削除、検索、および読み込みを行なう。コネクタデータ管理部920は、本実施の形態におけるコネクタ提供サーバ280がサービス利用端末100よりコネクタの要求を受けると、コネクタデータ保存領域900を検索して該当するコネクタを抽出する。該当コネクタデータ910を発見した場合は、前記コネクタデータ910に対して必要事項を読み込み、さらに必要に応じてコネクタデータ910のデータを編集して、通信制御部530に前記サービス利用端末100に対して前記コネクタデータ910の送信を依頼する。

【0110】

コネクタ提供サーバ280は、ネットワーク200を介して接続されたサービス提供サーバ210よりコネクタを取得してコネクタデータ保存領域900に保存する。その際の、コネクタ提供サーバ280におけるコネクタの受信処理について、図44のフローチャートを用いて説明する。

【0111】

図44を参照して、コネクタ提供サーバ280はステップS510において、サービス提供サーバ210より送信されるコネクタデータ910を受信する。

【0112】

ステップS510の受信が完了すると、続いて、ステップS511において、受信が正常に完了したかを判断する。受信が正常終了した場合（S511でYES）、ステップS512に進む。すなわち、受信が正常終了した場合には、受信したコネクタデータ910をコネクタデータ保存領域900に保存し、ステップS513に進む。

【0113】

ステップS513において、コネクタ提供サーバ280は、コネクタデータ910の提供元であるサービス提供サーバ210の各種情報をサービス提供サーバ情報テーブルとしてコネクタデータ管理部920に保存する。ステップS513でコネクタデータ管理部920に保存されるサービス提供サーバ情報テーブルの具体例を図45に示す。図45を参照して、サービス提供サーバ情報テーブルにおいては、会社名“AAA Corp.”より、ID“LS4369P3550”で名前“TextArt”的コネクタデータが取得されたことが示されている。また、入手日は2003年4月1日であり、当該コネクタ提供サーバ280は、残り100回、当該コネクタデータを配布してもいいことが示されている。

【0114】

ステップS513においてサービス提供サーバ210の情報の追加が完了すると、サービス提供サーバ210とコネクタ提供サーバ280との通信を終了し、同時に、コネクタ受信処理を終了する。

【0115】

一方、ステップS511においてコネクタデータの受信が異常終了した場合（S511でNO）、ステップS514において、受信データの消去やサービス提供サーバ210への受信失敗通知等のエラー処理を行ない、コネクタ受信処理を終了する。

【0116】

このようなコネクタデータ受信処理を実行して、コネクタ提供サーバ280は、サービス提供サーバ580から提供されるサービスにアクセス可能なコネクタデータをコネクタデータ管理部920に新たに保持することになる。

【0117】

なお、本実施の形態においては、サービス提供サーバ210からコネクタ提供サーバ280に直接通信を行なうことによってコネクタの転送を行なっているが、これは1つの具体例であり、他の方法として、通信を介さずにCD-ROM（Compact Disc-ROM）やメモリカード等の外部メモリメディアを利用して転送を行なってもよい。また、コネクタの提供元はサービス提供サーバ210に限定されず、提供するコネクタの作成を行なった図1には示されていない他の装置であってもよい。

【0118】

なお、コネクタデータ910の構成は、図9～図13、および図43に具体例が示されるコネクタデータ710の構成と同様であるため、ここでの説明を繰返さない。

【0119】

次に、本実施の形態におけるサービス利用端末100において当該サービス利用端末100に含まれるアプリケーションを実行する際に、コネクタを介して当該サービス利用端末100外の機能および当該サービス利用端末100内の機能を利用する場合の処理について説明する。ここでは、説明のために図1に示されたサービス提供システムの構成を簡略化し、図16に示される構成であるものとする。すなわち、図16を参照して、本具体例において、サービス利用端末1301は、画面を表示するディスプレイ1302と、サービス利用端末1301内の機能として、Camera機能1303と、File機能1304と、Send機能1305とを備える。また、サービス利用端末1301は、ネットワーク200を介してコネクタ提供サーバ1307と接続し、コネクタ提供サーバ1307が提供するコネクタを取得する。また、サービス利用端末1301は、ネットワーク200を介してサービス提供サーバ1306と接続し、サービス提供サーバ1306の提供するサービス利用端末1301外の機能であるTextArt機能を利用する。

【0120】

図17に、サービス利用端末1301においてアプリケーションを実行する際の処理の流れをフローチャートで示す。図17のフローチャートに示される処理は、サービス利用端末1301においてアプリケーションを実行する際に、制御部12が記憶部13に記憶されるプログラムを読み出して実行することで、図3に示される各部が制御されて実現される。なお、以降の説明においては、図6に示されるアプリケーションデータ310および図7に示されるアプリケーション405が実行されるものとして具体的に説明する。

【0121】

図17を参照して、始めに、ステップS101でアプリケーションが起動される。次にステップS102で、ディスプレイ1302に図18に具体例が示される入力画面が表示される。すなわち、ステップS102で制御部12は、図7に示されるアプリケーション405の記述601に設定される画面表示を実行する。

【0122】

図18を参照して、ステップS102でディスプレイ1302に表示される入力画面は、画像が表示されるImage領域1701と、文字が表示されるText領域1702と、TextArt機能を利用するためのTextArtボタン1703と、Camera機能1303を利用するためのCameraボタン1704と、File機能1304を利用するためのFileボタン1705と、Send機能1305を利用するためのSendボタン1706とを含んで構成され、それぞれの表示位置は、アプリケーション405の記述601に設定された位置である。

【0123】

ステップS102では、図18に示されるような入力画面においてユーザからの入力を受付け、その入力を判定する。そして、ステップS102での判定の結果、ユーザによる入力が当該サービス利用端末1301内外の機能を利用するものであれば（ステップS102で「機能利用」）ステップS1041に処理を進め、それ以外であれば（ステップS102で「機能利用以外」）ステップS103に処理を進める。

【0124】

ステップS102における判定方法としては、次のような方法が採用できる。すなわち、図7に示されたアプリケーション405において、ユーザの入力が記述603で設定されるevent要素のいずれかに対応し、そのevent要素が記述602で設定されるaction要素において各機能に対応付けられているかを解析することにより判定する。より具体的には、ステップS102では、図7に示されたアプリケーション405の場合、図18に表示されたボタン1703～1706のいずれかがユーザにより押下された場合、サービス利用端末1301内外の機能を利用する場合と判定する。

【0125】

次に、ステップS102での判定の結果、ユーザによる入力が当該サービス利用端末1301内外の機能を利用するもの以外である場合（S102で「機能利用以外」）、ステップS103では、端末内外の機能を利用しない場合のユーザの入力を処理してステップS102に処理を戻す。なお、ここでの処理は端末固有の処理など本発明において限定されない一般的な処理であって本発明の本質には関わらない処理であるため、ここでの説明は行なわない。

【0126】

一方、ステップS102での判定の結果、ユーザによる入力が当該サービス利用端末1301内外の機能を利用するものである場合（S102で「機能利用」）、さらにステップS1041で、ユーザの利用したい機能に対応するコネクタを検索する範囲を判定し、その中から所定のコネクタを検索する。ステップS1041の処理は、実行しているアプリケーション405に含まれる機能の動作を設定する記述に基づいて行なわれる。具体的には、実行しているアプリケーション405が図7に示されるものである場合、記述602のScope属性値を参照して、必要となるコネクタを規定される検索範囲から検索する。ここでは、「scope="Terminal"」と記述されているため、サービス利用端末1301内部よりfunc=" "で指定される条件に合致するコネクタを検索する。なお、ステップS1041での検索の結果、指定される条件に合致するコネクタが複数ある場合には（S1042でYES）、図示しない選択画面をディスプレイ1302に表示するなどして、ユーザから利用するコネクタの選択を受付けることが好ましい（S1043）。

【0127】

ステップS1041での検索の結果、指定される条件に合致するコネクタがサービス利用端末1301内にある場合には（S1044でYES）、処理をステップS1052に進め、サービス利用端末1301内部のコネクタデータ保存領域700から該当するコネクタデータを取得する。そして、処理をステップS108に進める。

【0128】

一方、ステップS1041での検索の結果、指定される条件に合致するコネクタがサービス利用端末1301内にない場合には（S1044でNO）、処理をステップS1051に進める。ステップS1051では、コネクタ提供サーバ1307に接続して必要なコネクタを検索し、サービス利用端末1301にダウンロードしてステップS106に処理を進める。

【0129】

ステップS106では、コネクタ提供サーバ280からコネクタを正しくダウンロードできたか否かを判定する。正しくダウンロードできた場合には（S106でYES）処理をステップS108に進め、ダウンロードに失敗した場合は（S106でNO）処理をス

ステップS107へ進め、エラーメッセージを生成して処理をステップS111にさらに進める。

【0130】

ステップS108においては、ステップS1052でサービス利用端末1301内から取得したコネクタ、またはステップS1051でコネクタ提供サーバ280からダウンロードしたコネクタを解析し、利用する機能がサービス利用端末1301内の機能であるか、サービス利用端末1301外の機能であるかを判定する。ステップS108での判定方法としては、コネクタデータ710の入出力定義805に基づいて判定する方法が挙げられる。より具体的には、コネクタデータ710の入出力定義805が図12に示される入出力定義805aである場合には、ステップS108において記述2503で設定されているその機能へのアクセス情報としてのアドレスURLより、その機能がサービス利用端末1301外の機能であると判定する(S108で「端末外の機能」)。一方、コネクタデータ710の入出力定義805が図12に示される入出力定義805bである場合には入出力定義805bにその機能へのアクセス情報としてアドレスURLを設定する記述2503が含まれないため、ステップS108においては、その機能がサービス利用端末1301内の機能であると判定する(S108で「端末内の機能」)。なお、上述のように、入出力定義805bにその機能へのアクセス情報として“Local”などを設定する記述2503が含まれている場合には、ステップS108においては、その記述2503に基づいてその機能がサービス利用端末1301内の機能であると判定する。

【0131】

そして、ステップS108での判定の結果、ユーザの利用したい機能がサービス利用端末1301外の機能である場合、すなわちサービス提供サーバ1306の提供する機能である場合は(S108で「端末外の機能」)、処理をステップS109に進め端末外機能利用処理を実行する。また、サービス利用端末1301内の機能である場合は(S108で「端末内の機能」)、処理をステップS110に進め端末内機能利用処理を実行する。なお、ステップS109における端末外機能利用処理、およびステップS110における端末内機能利用処理については、後にサブルーチンを挙げて詳細に説明する。

【0132】

そして、ステップS109またはステップS110の処理を終了すると、処理をステップS111に進める。すなわち、ステップS111では、ステップS109またはステップS110の処理の結果に応じてディスプレイ1302の画面表示を更新する。あるいは、ステップS107で生成されたエラーメッセージをディスプレイ1302へ表示する。そしてその後、処理をステップS102に戻して上述の処理を繰返す。

【0133】

さらに、図19に、上述のステップS109における端末外機能利用処理をフローチャートで示す。すなわち、図19を参照して、サービス利用端末1301においてサービス利用端末1301外の機能を利用する場合、すなわち、サービス提供サーバ1306の提供する機能を利用する場合、まずステップS201において、実行中のアプリケーションから、利用しようとしているサービス利用端末1301外の機能が利用可能であるかどうかを判定する。ステップS201での判定は、コネクタに設定された識別子と、アプリケーションに設定された識別子とを比較することにより行なわれる。具体的には、図10に具体例が示されたコネクタデータ710には、コネクタ識別子802において、vend or Limitという要素にAAA Coop. という値が設定されており、また図6に具体例が示されたアプリケーションデータ310には、アプリケーション識別子402において、vendorという要素にAAA Coop. という値が設定されている。ステップS201においては、これらが一致するため、図6に示されたアプリケーションから図10に示されたコネクタに設定された端末外の機能が利用可能であると判定される。

【0134】

このように、ステップS201において、実行しているアプリケーションから外部機能が利用可能と判定された場合は(S201でYES) 処理をステップS202に進め、利

用できないと判定された場合は（S201でNO）処理をステップS206に進める。

【0135】

ステップS202では、コネクタの有効期限と最大利用回数とをチェックして、そのコネクタがまだ利用可能であるかどうかを判定する。サービス利用端末1301において対象となるコネクタがそれまで何度利用されたかは記憶領域160に記憶されているものとし、現在の日時はサービス利用端末1301より取得できるとする。ステップS202において、コネクタの有効期限および最大利用回数は、図9に示すコネクタ710の有効期限803および最大利用回数804に設定されている値を参照することにより取得できる。

【0136】

ステップS202における判定の結果、コネクタの有効期限が切れておらず、かつこれまでの利用回数が最大利用回数に達していない場合は（S202でYES）、処理をステップS204に進め、有効期限が切れている、またはこれまでの利用回数が最大利用回数に達している場合は（S202でNO）、処理をステップS203に進める。

【0137】

ステップS203では、そのコネクタの利用ができない旨をディスプレイ1302に表示し、その後処理をメインルーチンのステップS1051に戻して、新たにコネクタをダウンロードする。

【0138】

一方、ステップS204では、サービス提供サーバ1306に接続して結果を得る。すなわち、ステップS204においては、外部機能管理部130の外部機能呼出部1101が、サービス提供サーバ1306に入出力定義805で定義されている必要なデータを引渡して接続する。そして、外部機能結果解析部1102がサービス提供サーバ1306から結果を受取り、実行しているアプリケーション405に引渡す。その後、ステップS204の処理が終了すると、処理をステップS205に進める。

【0139】

ステップS205では、サービス提供サーバ1306より正しい結果が得られたかどうかを判定する。ここでの判定は、たとえば、サービス提供サーバ1306より得たデータが入出力定義805で定義されている型のデータであるか否かを判定する、などの方法が挙げられる。また、そして、入出力定義805で定義されているアクセス情報でサービス提供サーバ1306に接続できたか否かを判定する方法も挙げられる。サービス提供サーバ1306に接続できない場合や、得られた結果が正しくない場合などは（S205でNO）処理をステップS206に進め、その旨を示すエラーメッセージを生成して本サブルーチンを終了し、処理を図17のメインルーチンへ戻す。また、正しく結果が得られた場合（S205でYES）は、その後本サブルーチンを終了して、処理を図17のメインルーチンへ戻す。

【0140】

以下、図20および図21に示される画面表示の具体例を用いて、外部機能を利用した際のサービス利用端末1301のディスプレイ1302に表示される画面遷移を説明する。

【0141】

図20に示される表示画面が表示されている状態において、サービス利用端末1301のText領域1702に「こんにちは」と入力され、TextArtボタン1703が押下されると「こんにちは」という文字列がTextArt機能を提供するサービス提供サーバ1306に渡される。そして、サービス提供サーバ1306より、当該TextArt機能を用いて装飾を施された「こんにちは」の文字列を得る。

【0142】

図21は、上記の処理により得られた、装飾された「こんにちは」の文字列がText領域1702に表示された状態を示す。これは、当該サービス利用端末1301において、図7に具体例が示されるアプリケーション405が実行されている場合、アプリケーシ

ヨン405の記述602に含まれる<setMedia func="name:TextArt" ... scope="Terminal" />の記述に基づいて図12に示される入出力定義805aを含むコネクタデータ710がコネクタデータ保存領域700から取得され、その入出力定義805aに、TextArt機能の存在場所としてサービス提供サーバ1306のアドレスを示す記述2503が含まれるためである。すなわち、サービス利用端末1301において実行されているアプリケーション405に含まれる機能に対応したコネクタデータ710の入出力定義805に、サービス提供サーバ1306のTextArt機能にはサービス利用端末1301のText領域に入力された文字列（ここでは「こんにちは」）を渡し、TextArt機能によって得られた結果をText領域1702に表示する旨が設定されているからである。

【0143】

次に、図22を用いて、図17のステップS110における端末内機能利用処理の詳細を説明する。

【0144】

図22を参照して、サービス利用端末1301において当該端末内の機能を利用する場合、まずステップS302において、実行中のアプリケーションから、利用しようとしているサービス利用端末1301内の機能が利用可能であるかどうかを判定する。ステップS302での判定は、コネクタに設定された識別子と、アプリケーションに設定された識別子とを比較することにより行なわれる。具体的には、図11に具体例が示されたコネクタデータ710には、コネクタ識別子802において、vendorLimitという要素にNoneという値が設定されているが、これはどのアプリケーションからもこのコネクタの機能を利用可能である、ということを意味する。したがって、この場合には、ステップS302においては、図6に示されたアプリケーションから図11に示されたコネクタに設定された端末内の機能が利用可能であると判定される。

【0145】

このように、ステップS302において、実行しているアプリケーションからサービス利用端末1301内部の機能が利用可能と判定された場合は（S302でYES）処理をステップS303に進め、利用できないと判定された場合は（S302でNO）処理をステップS306に進める。

【0146】

ステップS303では、コネクタの有効期限と最大利用回数とをチェックして、そのコネクタがまだ利用可能であるかどうかを判定する。当該サービス利用端末1301において対象となるコネクタがそれまで何度利用されたかは記憶領域160に記憶されているものとし、現在の日時はサービス利用端末1301より取得できるとする。ステップS303において、コネクタの有効期限および最大利用回数は、図9に示すコネクタ710の有効期限803および最大利用回数804に設定されている値を参照することにより取得できる。

【0147】

ステップS303における判定の結果、コネクタの有効期限が切れておらず、かつこれまでの利用回数が最大利用回数に達していない場合は（S303でYES）、処理をステップS304に進め、有効期限が切れている、またはこれまでの利用回数が最大利用回数に達している場合は（S303でNO）、処理をステップS306に進める。具体的には、図11に具体例が示されるコネクタデータ710の場合、有効期限803にはexpireDateがNoneと設定されており、これは有効期限が未設定すなわち常に利用可能を意味する。また、有効利用回数804にはexpireNumberがNoneと設定されており、これは最大利用回数が未設定すなわち何度でも利用可能であることを意味する。

【0148】

ステップS304では、サービス利用端末1301の備える内部機能を起動して結果を得る。すなわち、ステップS304においては、内部機能管理部140の内部機能呼出部

1201が、サービス提供サーバ1306に入出力定義805で定義されているサービス利用端末1301内部の機能に必要なデータを引渡して起動させる。そして、内部機能結果解析部1202が内部機能からの結果を受取り、実行しているアプリケーション405に引渡す。その後、ステップS304の処理が終了すると、処理をステップS305に進める。

【0149】

ステップS305では、起動したサービス利用端末1301内部の機能より正しい結果が得られたかどうかを判定する。ここでの判定は、たとえば、サービス利用端末1301内部の機能より得たデータが入出力定義805で定義されている型のデータであるか否かを判定する、などの方法が挙げられる。判定の結果、得られた結果が正しくない場合は(S305でNO)処理をステップS306に進め、その旨を示すエラーメッセージを生成して本サブルーチンを終了し、処理を図17のメインルーチンへ戻す。また、正しく結果が得られた場合は(S305でYES)は、その後本サブルーチンを終了して、処理を図17のメインルーチンへ戻す。

【0150】

以下、図21、図23～図27に示される画面表示の具体例を用いて、内部機能を利用した際のサービス利用端末1301のディスプレイ1302に表示される画面遷移を説明する。

【0151】

図21に示される表示画面が表示されている状態において、サービス利用端末1301のCameraボタン1704が押下されると、当該端末内のCamera機能1303が起動される。

【0152】

図23は、Camera機能1303が起動した状態のサービス利用端末1301のディスプレイ1302に表示される撮影画面の具体例を示す図である。図23を参照して、撮影画面は、図示しないカメラレンズの捕らえている画像が表示される領域2001と、シャッタボタン2002とを含む。

【0153】

図23に示される撮影画面が表示されている状態においてシャッタボタン2002が押下されると、ディスプレイ1302に表示される画面は図24に示される合成画面に遷移する。図24を参照して、合成画面のImage領域1701にはCamera機能1303により撮影された画像が表示されている。これは、サービス利用端末1301において、たとえば図7に示されるアプリケーション405が実行されている場合、アプリケーション405の記述602に含まれる<setMedia func="name:Camera" ... scope="Terminal" />の記述に基づいて図13に示される入出力定義805bを含むコネクタデータ710がコネクタデータ保存領域700から取得され、その入出力定義805bに、サービス提供サーバ1306内部のCamera機能1303により撮影された画像をImage領域1701に表示する旨が設定されているからである。

【0154】

さらに図24に示される合成画面が表示されている状態においてFileボタン1705が押下されると、サービス利用端末1301内のFile機能1304が起動し、ディスプレイ1302に表示される画面は図25に示されるファイル選択画面に遷移する。図25を参照して、ファイル選択画面は、当該端末の記憶領域160に記憶されているファイルリストが表示されるFile領域2201と、OKボタン2202とを含む。

【0155】

図25に示されるファイル選択画面が表示されている状態においてFile領域2201から所定のファイルが選択されOKボタン2202が押下されると、記憶領域160から選択されたファイルが読み出され、ディスプレイ1302に表示される画面は図26に示されるファイル表示画面に遷移する。

【0156】

図26を参照して、ファイル表示画面のImage領域1701にはFile機能1304により選択された画像が表示されている。これは、当該サービス利用端末1301において、たとえば図7に示されるアプリケーション405が実行されている場合、アプリケーション405の記述602に含まれる<setMedia func="name:File" ... scope="Terminal" />の記述に基づいて、その入出力定義805に、File機能の存在場所としてサービス提供サーバ1306のアドレスを示す記述2503が含まれていない、または当該サービス利用端末1301を示す記述2503が含まれているコネクタデータ710がコネクタデータ保存領域700から取得され、その入出力定義805に、サービス利用端末1301のFile機能1304により選択された画像をImage領域1701に表示する旨が設定されているからである。

【0157】

さらに図26に示されるファイル表示画面が表示されている状態においてSendボタン1706が押下されると、サービス利用端末1301内のSend機能1305が起動される。Send機能1305が起動されることで、これまでの説明により作成されたコンテンツ(ImageおよびText)が予め指定されていた宛先に送信される。図27には、送信が完了したときにディスプレイ1302に表示される送信完了画面の具体例であって、送信が完了した旨を表わす画面の具体例である。図27に示されるように、Send機能1305が起動されて作成されたメッセージが送信されると、その旨を示す送信完了表示2203がディスプレイ1302に表示される。

【0158】

このように、本発明にかかるサービス利用端末は図28に示されるようなサービス利用処理を実行してコネクタを介して機能を利用するによって、サービスの提供を享受することができる。すなわち、図28を参照して、携帯電話などのサービス利用端末において、ユーザ入力を受付けてアプリケーションを実行する際に、アプリケーションに必要な機能をコネクタを介して利用する。この際、必要に応じて、コネクタをコネクタ提供サーバに要求して取得する。このようにすることで、サービス利用端末の内部の機能と当該端末の外部のサービス提供サーバが提供する機能とを同一の方法でアプリケーションにおいて利用することができる。すなわち、アプリケーションを利用するユーザは、そのアプリケーションに必要な機能がそのサービス利用端末が備えられているか否かを意識することなく利用することができ、ユーザの利便性を高めることができる。

【0159】

さらに、特定の機能がアプリケーションから利用可能であるか否かを、コネクタの識別子と実行しているアプリケーションの識別子とを比較(マッチング)して判断することによって、サービス利用端末内外の機能を利用できるアプリケーションを制限することができる。また、サービス提供サーバの提供する機能をコネクタを介して利用することにより、コネクタの提供されていないサービス提供サーバの提供する機能の利用を制限することができる。すなわち、アプリケーション作成者の意図しない機能が用いられるこれを防止することができる。具体的には、アプリケーション実行時に、その機能を利用するためアプリケーション作成者が意図していないコネクタが用いられるという事態を防止することができる。その結果、アプリケーション作成者が意図していない機能がアプリケーションの実行に用いられるという事態を防止することができ、サービス利用端末においてアプリケーション作成者の意図する実行が行なわれる。また、たとえばサービス利用端末に有害な機能がサービス利用端末において実行されてしまうという事態を防止することができる。このため、Webサービスを利用するユーザの安全性を高めることができる。

【0160】

さらに、アプリケーションを実行する際に必要な機能の存在場所をコネクタで設定することによって、アプリケーション作成者は、必要な機能がサービス利用端末に備えられて

いるか否か、また、サービス提供サーバに備えられているか否かを意識することなく、コネクタを設定することで意図する機能が実行されるアプリケーションを作成することができる。そのため、アプリケーションを実行する際に必要な機能の存在場所やバージョンなど、その機能の情報が変更された場合であっても、その変更をアプリケーションに反映する必要がなく、コネクタを書換える、またはコネクタを置換えるだけでその変更をアプリケーションに反映することが可能であるため、アプリケーションの作成が容易になる。

【0161】

【変形例1】

第1の変形例として、アプリケーション405の他の具体例を図29に示す。第1の変形例においては、アプリケーション405において、実行時に用いられる機能に対応したコネクタが、記述6021のような検索条件で設定されることを特徴としている。すなわち、図29を参照して、具体的に第1の変形例においてアプリケーション405は、TextArtというアクションの動作を、「検索範囲を“Terminal+Server（サービス利用端末100およびコネクタ提供サーバ280）”として検索された、入力するデータの型が文字列で、出力するデータの型がJpeg画像であるコネクタを利用し、該当する機能にはText領域の値を渡し、その機能から受取った結果をText領域に表示する」と設定している。

【0162】

図29に示されるようなアプリケーション405をサービス利用端末1301において実行する際、上述のステップS104においては、コネクタの検索範囲をサービス利用端末1301内外の両方と判定し、ステップS1051および／あるいはステップS1052において該当するコネクタを取得する。具体的には、図12に示されるようなコネクタデータ710の入出力定義805の記述2501および記述2502を参照して、そのコネクタに対応している機能に対して入力するデータの型が文字列であって、そのコネクタに対応している機能が出力するデータの型がJpeg画像であるコネクタを検索する。

【0163】

なお、ステップS1051および／あるいはステップS1052において該当するコネクタが複数得られた場合、ステップS108の処理に先だって、図30に示されるようなコネクタ選択画面2204をディスプレイ1302に表示し、利用するコネクタを選択することが好ましい。

【0164】

アプリケーションにおいてこのような検索条件でコネクタを設定することで、アプリケーション作成者がコネクタの存在場所を知らない場合であっても、意図する機能を実行させるためのコネクタを設定することが可能である。

【0165】

また、サービス利用端末においてこのようなアプリケーションを実行する際には、検索された複数のコネクタの中から、最適なコネクタを選択してそのコネクタに対応した機能を実行することができる。

【0166】

【変形例2】

第2の変形例として、アプリケーション405のさらに他の具体例を図31に示す。第2の変形例においては、アプリケーション405において、実行時に用いられる機能に対応したコネクタが、記述6022のような検索条件で設定されることを特徴としている。すなわち、図31を参照して、具体的に第2の変形例においてアプリケーション405は、TextArtというアクションの動作を、「検索範囲を“Terminal+Server（サービス利用端末100およびコネクタ提供サーバ280）”として検索された、サービスの記述が“文字列装飾”であるコネクタを利用し、該当する機能にはText領域の値を渡し、その機能から受取った結果をText領域に表示する」と設定している。

【0167】

図31に示されるようなアプリケーション405をサービス利用端末1301において

実行する際、上述のステップS104においては、コネクタの検索範囲をサービス利用端末1301内外の両方と判定し、ステップS1051および／あるいはステップS1052において該当するコネクタを取得する。具体的には、図12に示されるようなコネクタデータ710の入出力定義805の記述2503などを参照して、（図12には図示されていない）サービスの記述に該当する記載のあるコネクタを検索する。

【0168】

なお、ステップS1051および／あるいはステップS1052において該当するコネクタが複数得られた場合、ステップS108の処理に先だって、図32に示されるようなコネクタ選択画面2205をディスプレイ1302に表示し、利用するコネクタを選択することが好ましい。

【0169】

アプリケーションにおいてこのような検索条件でコネクタを設定することでも、第1の変形例と同様に、アプリケーション作成者がコネクタの存在場所を知らない場合であっても、意図する機能を実行させるためのコネクタを設定することが可能である。

【0170】

また、サービス利用端末においてこのようなアプリケーションを実行する際には、検索された複数のコネクタの中から、最適なコネクタを選択してそのコネクタに対応した機能を実行することができる。

【0171】

【変形例3】

第3の変形例として、コネクタデータ710の他の具体例を図33に示す。第3の変形例においては、コネクタデータ710において、対応する機能で扱うデータの種類を考慮して、記述806のようにアクセスの制限が設定されることを特徴としている。すなわち、図33を参照して、具体的に第3の変形例においてコネクタデータ710は、File機能が扱うデータのうち、システムデータは読み専用のデータであって表示せず、住所録データは読み専用のデータであって表示し、一般データは読み書き可能のデータであって表示する、と設定している。

【0172】

サービス利用端末1301において図33に示されるようなコネクタデータ710を介してFile機能を実行する際、ステップS111における結果表示では、図34に示すようなファイル選択画面を表示する。すなわち、図34を参照して、File領域2206にはFile機能で扱うデータとしてSample1.jpg、Sample2.jpg、およびSample3.jpgを[RW]と表示し、読み書き（削除）が可能なデータである旨を示している。また、住所録1.add、および住所録2.addを[RO]と表示し、閲覧のみが可能であって内容の変更や削除ができないデータである旨を示している。さらに、prop="Hidden"と表示を制限されているファイル（図示せず）を表示していない。

【0173】

このように、コネクタに対応する機能が扱うデータへのアクセスを制限することによって、よりセキュリティの高いサービスの提供を行なうことができる。

【0174】

なお、言うまでもなく、このような対応する機能が扱うデータへのアクセスの制限はコネクタに設定される場合に限定されず、コネクタに対応している機能によって実現されてもよい。

【0175】

【変形例4】

第4の変形例におけるサービス提供システムの、簡略化した構成の具体例を図35に示す。図35に示されるように、第4の変形例においては、サービス利用端末100において、サービス提供サーバ210から当該サービス利用端末100に応じてカスタマイズされた機能であるアプリケーションの提供、また、コネクタ提供サーバ280から当該サー

ビス利用端末100に応じてカスタマイズされたコネクタの提供を享受することを特徴とする。

【0176】

具体的には、サービス利用端末100がサービス提供サーバ210に対して特定の機能を要求したとき、サービス提供サーバ210においてその機能がそのサービス利用端末100に応じてカスタマイズされ、アプリケーションデータ310に図36に示されるようなサービス利用端末100に応じたアプリケーション識別子402が付与される。

【0177】

また、サービス利用端末100がコネクタ提供サーバ280に対して特定のコネクタを要求したとき、コネクタ提供サーバ280においてそのコネクタがそのサービス利用端末100に応じてカスタマイズされ、コネクタデータ710に図37に示されるようなサービス利用端末100に応じたコネクタ識別子802が付与される。

【0178】

このようなアプリケーション識別子402およびコネクタ識別子802は、サービス利用端末100から要求を受けたサービス提供サーバ210およびコネクタ提供サーバ280において一意に割振られるものであって、サービス利用端末100の記憶部13の一般ユーザ変更不可領域132に記憶されている固有情報などを用いた識別子である。このようなサービス利用端末100の固有情報は、要求と共にサービス利用端末100から自動的に送られてくるものであってもよいし、要求を受けたサービス提供サーバ210およびコネクタ提供サーバ280が当該サービス利用端末100などから取得するものであってもよい。また、サービス提供サーバ210およびコネクタ提供サーバ280がアクセス可能なデータベース（図示せず）に予め格納されている情報であってもよい。

【0179】

このような場合、サービス利用端末100においては、図38に示されるような端末外機能利用処理が実行される。すなわち、ステップS401において図19のステップS201と同様の処理が実行されて、実行しているアプリケーションから外部機能が利用可能と判定された場合（S401でYES）、処理をステップS412に進めて、そのアプリケーションの識別子402とコネクタの識別子802とを比較し、当該サービス利用端末100においてそのアプリケーション405を実行する際にそのコネクタ710を介して対応付けられている機能を利用できるか否かを判定する。ステップS412では、識別子402、802が共に、サービス利用端末100に応じてカスタマイズされた識別子であることを確認する。具体的には、ステップS412では、両識別子402、802に当該サービス利用端末100を示す情報が含まれているか否か、などを確認する。

【0180】

ステップS412での比較の結果、アプリケーションの識別子402とコネクタの識別子802とがサービス利用端末100に応じてカスタマイズされた識別子であることが確認された場合には（S412でYES）、処理をステップS402へ進める。ステップS402以降の処理は、上述のステップS202以降の処理と同様であるため、ここでの説明は繰返さない。

【0181】

一方、ステップS412での比較の結果、アプリケーションの識別子402とコネクタの識別子802との少なくとも一方がサービス利用端末100に応じてカスタマイズされた識別子でないことが確認された場合には（S412でNO）、ステップS406へ処理を進め、エラーメッセージを生成して本処理を終了する。そして、処理を図17のメインルーチンへ戻す。

【0182】

このように、第4の変形例におけるサービス利用端末では、サービス利用端末に応じてカスタマイズされたアプリケーションとコネクタとを用いてアプリケーションを実行することを特徴とする。このことによって、サービス利用端末において実行されるアプリケーションとコネクタとのセキュリティが確実に保たれる。また、アプリケーションとコネク

タとのマッチングが不成功の場合には、そのアプリケーションにおいて対応しないコネクタを用いることができないため、サービス利用端末においてアプリケーション作成者の意図する機能を実行することができる。

【0183】

ところで、上述の図17に示されるサービス利用端末1301での処理において、実行しているアプリケーション405が図46に示されるものである場合、ステップS1041では、記述6023のScope属性値を参照して、必要となるコネクタを規定される検索範囲から検索する。ここでは、「scope = "Server"」と記述されているため、ステップS1044でコネクタ提供サーバ1307に接続して必要となるコネクタをダウンロードして取得するものと判断する。または、アプリケーション405のScope属性値が「scope = "Terminal+Server"」と記述されている場合には、サービス利用端末1301は、必要となるコネクタを、まず自端末のコネクタデータ保存領域700から検索し、そこで得られなかった場合に、次に、コネクタ提供サーバ1307に接続して必要となるコネクタをダウンロードして取得するものと判断する。または、Scope属性値に関わらず、コネクタが必要となるたびにコネクタ提供サーバ1307に接続してダウンロードしてもよいし、コネクタ提供サーバ1307が提供するコネクタと自端末のコネクタデータ保存領域700に保持するコネクタとを比較して、コネクタ提供サーバ1307が提供するコネクタが更新されている場合にコネクタ提供サーバ1307に接続して必要となるコネクタをダウンロードして取得してもよい。

【0184】

ステップS1051では、コネクタ提供サーバ1307に対して必要とするコネクタを要求する。なお、上述の図46は、アプリケーション405が拡張されたSMILにより記述されている場合の具体例であるが、アプリケーション405は、Java(R)言語やC言語を始めとする他のプログラムでの実装や、または、他の書式によって表現されていてもかまわない。

【0185】

この場合の、コネクタ提供サーバ1307におけるコネクタ送信処理について、図47のフローチャートを用いて説明する。図47のフローチャートに示される処理は、コネクタ提供サーバ1307においてコネクタをサービス利用端末1301に提供するサービスを実行する際に、制御部22が記憶部23に記憶されるプログラムを読み出して実行することで、図41に示される各部が制御されて実現される。

【0186】

図47を参照して、始めに、ステップS601でサービスが起動される。次にステップS602で、サービス利用端末1301からコネクタの要求があると(S602でYES)、ステップS603でコネクタデータ管理部920において要求元のサービス利用端末1301の情報を取得して確認する。

【0187】

ステップS603では、コネクタ要求を示す依頼情報自体に含まれる端末IDをサービス利用端末情報として取得してサービス利用端末1301を確認してもよいし、また、サービス利用端末1301のユーザID情報をサービス利用端末情報として利用してもよい。また、サービス利用端末情報として、サービス利用端末情報に基づいてそのコネクタの提供が許可されているか否かを判断できる他のいづれの情報を利用してもよい。また、さらに詳しい情報を取得するために、ステップS603において、コネクタ提供サーバ1307は、要求のあったサービス利用端末1301に対して、端末自身の情報や当該端末を利用しているユーザ情報の問い合わせを行なってもよい。

【0188】

次に、ステップS604では、ステップS603で取得したサービス利用端末情報に基づいて、サービス利用端末1301に対してコネクタを提供するか否かを判断する。ステップS604においてコネクタを提供しないと判断した場合はステップS610に進む。また、ステップS604における判断結果においてコネクタを提供すると判断した場合は

、ステップS605に進む。

【0189】

ステップS604における判断は、ステップS603で取得したサービス利用端末情報に基づいて行なわれてもよい。より具体的には、予め端末管理部510に記憶されているコネクタ提供が可能な端末の情報を参照して、ステップS603で取得したサービス利用端末情報に基づいて判断してもよい。また、ステップS603で取得したサービス利用端末情報に基づいて他の認証サーバへの問合わせを行なった結果に基づいて行なわれてもよい。

【0190】

なお、コネクタ提供サーバ1307において、サービス利用端末1301のいずれのサービス利用端末情報も利用しない場合は、前述のステップS603を省いてステップS604に進んでもよい。この場合、ステップS604では、サービス利用端末1301からの情報を利用せずに、コネクタ提供サーバ1307自体の状態に応じてコネクタを提供するか否かが判断される。また、コネクタ提供サーバ1307において、いずれのアクセス制限も行なわない場合は、前述のステップS604を省いて、ステップS605に進んでもよい。

【0191】

ステップS605では、コネクタ管理部920において、要求されたコネクタがコネクタデータ保存領域900に保持されているかどうかの確認を行なう。要求されたコネクタがコネクタデータ保存領域900に保持されているか否かは、コネクタデータ管理部920において、該当するコネクタがあるか否かコネクタデータ保存領域900の検索を行うことによって判断される。

【0192】

コネクタ提供サーバ1307が該当するコネクタを所有していない場合（S605でNO）、ステップS610のエラー通知処理に進む。一方、コネクタ提供サーバ1307が該当するコネクタを所有している場合（S605でYES）、ステップS606に進み、コネクタデータ管理部920において必要に応じて所有しているコネクタデータ910のデータの一部を編集した後ステップS607に進み、通信制御部530にて、サービス利用端末1301に向けて該当コネクタデータ910を送信する。

【0193】

ステップS606でのコネクタの編集の1つの具体例としては、ステップS603において取得したサービス利用端末1301の端末情報に応じてコネクタデータ910を編集する方法が挙げられる。

【0194】

より具体的には、サービス利用端末1301の端末情報を参照することにより、サービス利用端末1301が課金済みのアプリケーションを起動していることを検知した場合、ステップS606でコネクタデータ管理部920はコネクタデータ910の最大利用回数を指定しない編集、または有効期限を無期限に指定する編集を行なう。このような編集を行なうことにより、サービス利用端末1301上で当該アプリケーションを実行する際、前述コネクタデータ910で特定される機能を無制限に利用することができる。

【0195】

またより具体的には、サービス利用端末1301の端末情報を参照することにより、サービス利用端末1301が「体験版」の状態でアプリケーションを起動していることを検知した場合、ステップS606でコネクタデータ管理部920はコネクタデータ310の最大利用回数を10回に指定する編集、または有効期限をリクエストのあった日から1週間後までの日付に指定する編集を行なう。このような編集を行なうことにより、サービス利用端末1301上で当該アプリケーションを実行する際、前述コネクタデータ910で特定される機能の利用回数や利用期間を制限することができる。

【0196】

またより具体的には、ステップS603において取得したサービス利用端末1301の

端末情報よりサービス利用端末1301の機種や契約している通信キャリアなどを特定し、ステップS606でコネクタデータ管理部920はそれらに応じてコネクタデータ910の編集を行なう。たとえば、特定されたサービス利用端末1301の機種が備える表示装置の解像度や、表示可能な色調や、特定されたサービス利用端末1301の通信キャリアにおいて通信可能なデータ容量などに応じて、コネクタデータ910で特定される機能から出力されるデータ形式を変更するなどの編集を行なうことが好ましい。

【0197】

また、コネクタの編集の他の具体例としては、サービス提供サーバ1306の形態に応じてコネクタデータ910を編集する方法が挙げられる。より具体的には、図48に示されるように、ある同一の機能を提供するサービス提供サーバ1306が複数台（Server1～Server3）存在する場合がある。これは、複数のサービス利用端末から同時にアクセスされた場合でも十分に対応できるように、同様の機能を提供するサービス提供サーバを複数台稼動させて、サービス利用端末からのリクエストごとに異なるサービス提供サーバが動作するような仕組みが導入された場合の一般的なシステム構成である。この場合、ステップS606でコネクタデータ管理部920は上記複数のサーバにランダムに、または所定のアルゴリズムに従ってコネクタデータ910に記載されるサービス提供サーバ名（サービス提供サーバへのアクセス情報）を変更する編集を行なう。具体的には、コネクタデータ910のサービス定義を図49の入出力定義の具体例である入出力定義805cの行2703のように編集する。このような編集を行なうことにより、サービス提供サーバにアクセスを分散させる仕組みを導入することなく、サービス利用端末1301から特定のサービス提供サーバにアクセスが集中することを防ぎ、アクセスを分散させることができる。

【0198】

また、サービス提供サーバ1306の形態に応じてコネクタデータ910を編集する方法として次のような他の具体例が挙げられる。すなわち、図50に示されるように、サービス提供システムがサービス利用端末1301とサービス提供サーバ1306との間で機能の提供を中継する中継サーバ（プロキシサーバとも言われる）1308を含み、サービス提供サーバ1306が中継サーバ1308を介してサービスを提供する形態である場合、ステップS606でコネクタデータ管理部920は、実際にサービスを提供しているサービス提供サーバ1306へのアクセス情報を、サービス中継サーバ1308へのアクセス情報に変更する編集を行なう。具体的には、コネクタデータ910のサービス定義を図51の入出力定義の具体例である入出力定義805dの行2803のように編集する。このような編集がされたコネクタを受信したサービス利用端末1301ではコネクタの記述に従って、サービス中継サーバ1308をサービス提供サーバ1306と仮定して機能の提供を要求する。一般的に、セキュリティ確保の観点より、携帯電話などのサービス利用端末において1つのアプリケーションを実行している際には、そのアプリケーションの実行においてアクセスできるサーバは1つに限定されている場合が多い。しかし、サービス提供サーバで提供される機能は複数のサーバから提供されている場合や、他のサーバが提供している同様の機能であってよりよい機能を利用したい場合もある。そのような場合に、コネクタの記述に従ってサービス中継サーバ1308をサービス提供サーバ1306と仮定してアクセスすることで、サービス利用端末1301では、サービス提供サーバへのアクセスの制限に問わらずに最適な機能を利用することができる。

【0199】

そして、ステップS608において送信処理が正常に終了した場合（S608でYES）、ステップS609で端末管理情報の更新処理を行なう。ステップS609では、端末情報管理部210において、ステップS603にて取得したコネクタを送信したサービス利用端末1301の情報と、当該端末に対してコネクタを送信した旨とが記録される。本処理においては、単にサービス利用端末1301の情報を記憶しておくだけでなく、図示しない他の課金サーバと連携して、コネクタ送信の記録と共にコネクタ発行に対する課金処理を行なってもよい。また、更新処理を不要とする場合は、ステップS608を省略し

て次に進んでもよい。

【0200】

一方、ステップS608において送信処理が異常終了したことが判断された場合（S608でNO）、ステップS610のエラー通知処理に進む。

【0201】

ステップS610では通信制御230において、コネクタを要求したサービス利用端末1301に対して、コネクタの要求手続きが異常終了したことの通知を行なう。ステップS610では、サービス利用端末1301自身がタイムアウト処理機能や通信エラー検知機能を備える場合にはエラー通知処理を省略して、コネクタ提供サーバ1307自身のエラー処理のみを行なってもよい。その後、処理をステップS602に戻し、次のコネクタ要求があるまで待機する。

【0202】

上述の処理を実行したコネクタ提供サーバ1307より要求するコネクタを受信したサービス利用端末1301は、受信したコネクタに記載されているアクセス情報を参照してサービス提供サーバ1306へアクセスし、サービス提供サーバ1306の提供する機能を利用できる。

【0203】

【変形例5】

第5の変形例として、上述の図50に示されるようなサービス提供システムにおいて、コネクタ提供サーバ1307が中継サーバ1308の機能を備えて、コネクタ提供サーバ1307と中継サーバ1308とが一体の構成となっていてもよい。

【0204】

その場合のコネクタ提供サーバ1307で実行されるサービス中継処理について図52のフローチャートを用いて説明する。図52のフローチャートに示される処理は、中継サーバとして機能するコネクタ提供サーバ1307がサービス利用端末1301からサービス提供サーバ1306に替わってサービスの提供の要求を受付けた際に制御部22が記憶部23に記憶されるプログラムを読出して実行することで、図41に示される各部が制御されて実現される。

【0205】

図52を参照して、始めに、ステップS700でサービス利用端末1301からのサービス提供の要求を受信したコネクタ提供サーバ1307は、ステップS701において、要求されたサービス（機能）が中継処理を行なう対象であるかの検索を行なう。ステップS701では、コネクタデータ管理部920は、サービス提供サーバ1306へのアクセス情報（アドレス）を中継サーバ1308として機能するコネクタ提供サーバ1307自身へのアクセス情報に書換える編集を行なったコネクタデータ910をサービス利用端末1301に提供したか否かを、コネクタデータ保存領域900に格納されているコネクタデータ910を検索して判定する。

【0206】

ステップS701での検索の結果、コネクタ提供サーバ1307自身へのアクセス情報に編集したコネクタデータ910をサービス利用端末1301に提供していた場合は、そのコネクタに記載されているサービス名を参照してコネクタデータ保存領域900を検索して、編集したコネクタデータ910に対応するオリジナルのコネクタデータを検索する。そして、コネクタデータ保存領域900よりオリジナルのコネクタデータを検出した場合は、サービス提供の要求を中継可能だと判断し（S702でYES）、ステップS703に進む。

【0207】

ステップS703では、サービス利用端末1301から受取ったサービス提供の要求を、オリジナルのコネクタデータに記載されているアクセス情報に基づいて該当するサービス提供サーバ1306に送る。続いて、ステップS704にて、サービス提供サーバ1306からのサービス応答を待つ。

【0208】

サービス提供サーバ1306から正常に結果が返ってきた場合（S704でYES）、ステップS705に進んで結果をサービス利用端末1301に送信し、サービス中継処理を終了する。

【0209】

一方、ステップS704でサービス提供サーバ1306から応答がなかった場合（S704でNO）や、ステップS702で要求されたサービス（機能）が中継処理を行なう対象でないと判断した場合（S702でNO）は、ステップS707に進んで受信データや一時保存データを破棄し、サービス提供を要求したサービス利用端末1301に対して、当該サービスが利用できない旨の通知を行なう等のエラー処理を行なう。

【0210】

さらに、ステップS702において、予めサービス提供サーバ1306がそのサービスの提供ができないことがわかつておりその旨をコネクタ提供サーバ1307に通知している場合は、コネクタ提供サーバ1307は当該サービスが利用できない予約がされていると判断し（S702で「処理の予約がされている場合」）、ステップS706で、予め設定されている（固定の）サービスが利用できない旨を示す情報をサービス利用端末1301に対して送信する。具体的には、サービス提供サーバがメンテナンス作業のため一定期間、サービスの提供ができない場合、一般的には、サービスの利用者に前もってサービスが停止する旨を通知したり、または、同じサービスを提供可能な代替サービス提供サーバを同じアドレスで立ち上げたりする必要があったが、本変形例では、前述の対応をとることなく、コネクタ提供サーバに代替サービス提供サーバの働きをさせることで、サービス提供サーバの構成が簡単になる。

【0211】

以上で、第5の変形例のコネクタ提供サーバにおけるサービス中継処理が終了する。

【0212】

【変形例6】

第6の変形例にかかるアプリケーション405の具体例を図53に示す。第6の変形例にかかるアプリケーション405ではevent要素に対して複数のaction要素が関連付けられており、ユーザの単一の入力操作により複数の機能を同時に実行することを特徴とする。

【0213】

具体的には、図53の記述5301において、「auto-f11」ボタンを押された場合に実行される“auto-f11”というアクションには3つの動作が割当てられており、それぞれ「検索範囲を“Terminal”として検索されたgetAddressという名前のコネクタを利用し、該当する機能には値を渡す必要はなく、その機能から受取った結果は“Text:1”領域に表示する」、「検索範囲を“Terminal”として検索されたgetCompanyという名前のコネクタを利用し、該当する機能には値を渡す必要はなく、その機能から受取った結果は“Text:2”領域に表示する」、および「検索範囲を“Terminal”として検索されたgetNameという名前のコネクタを利用し、該当する機能には値を渡す必要はなく、その機能から受取った結果は“Text:3”領域に表示する」という動作を設定している。

【0214】

本具体例にかかるアプリケーションにおいてこのようにコネクタを利用することで、図54に示される入力画面のような、たとえば、住所、会社名、および名前など一連の個人情報を入力する必要がある場合に、その情報が端末内または端末外のどちらにあるかに関わらず、「auto-f11」ボタンを押すという単一の操作により図55に示されるようにそれらの情報が自動的に入力される。このため、ユーザの手間を省くことができる。

【0215】

【変形例7】

第7の変形例にかかるアプリケーション405の具体例を図56に示す。第7の変形例にかかるアプリケーション405では、文字列を表示するtext要素が、複数の要素から条件に合致した要素を選出すswitch要素の対象となっており、この条件がコネクタから受取った結果に基づいているものであることを特徴とする。

【0216】

具体的には、図56の記述5601においてswitch要素には2つのtext要素が割当てられており、それぞれ、「検索範囲を“Terminal”として検索されたgetRoleという名前のコネクタを利用し、その機能から受取った結果が“chief”もしくは“director”的に、「Text」領域にファイル“secret.txt”的な内容を表示する」、および「検索範囲を“Terminal”として検索されたgetRoleという名前のコネクタを利用し、その機能から受取った結果が空文字の場合に、「Text」領域にファイル“public.txt”的な内容を表示する」という動作を設定している。

【0217】

本具体例にかかるアプリケーションにおいてこのようにコネクタを利用することで、たとえば企業向けのワークフローなどを想定した場合に、同じアプリケーションでも重要な情報を見る権限を持つ社員に対しては図57に示されるように表示し、一般の社員に対しては図58に示されるように表示するといったように、権限によって情報の見せ方を変える、つまり表示の範囲を所定の条件に応じて変化させることが可能である。

【0218】

また、図59に示されるアプリケーション405のように、ボタン表示の部分に同様なswitch要素を割当てることにより、図60および図61に示されるように、ボタン（「approve」ボタン）そのものを表示・非表示することができ、たとえば決裁権限を持つ社員にのみ決裁書に「決裁」ボタンを表示するといった表示方法を行なうことが可能である。

【0219】

【変形例8】

第8の変形例にかかるコネクタデータの保存方法および管理方法の具体例を図62に示す。第8の変形例においては、図62に示されるように、コネクタデータ保存領域およびコネクタデータ管理部がネットワークによって接続されたサーバに保持されて、サービス利用端末1031内のコネクタ管理部1031には、該サーバに保持されるコネクタデータ保存領域およびコネクタデータ管理部の存在場所のみが保持されることを特徴とする。

【0220】

この場合、上述のステップS1044においては、サーバに保持されたコネクタデータ保存領域が検索されて、該コネクタデータ保存領域に該当するコネクタデータが存在すれば、ステップS1052にてコネクタデータ保存領域より該当するコネクタが取得される。

【0221】

一方、該当するコネクタデータが見つからない場合は、ステップS1051にてコネクタ提供サーバ280より該当するコネクタがダウンロードされ、サービス利用端末1031のコネクタデータ保存領域に保存される。複数のサービス利用端末についてのコネクタデータ保存領域およびコネクタデータ管理部を保持するサーバは、サービス利用端末ごとのコネクタデータ保存領域およびコネクタデータ管理部を当該サービス利用端末に関連付けて保持する。

【0222】

本具体例においては、以上のような構成により、各サービス利用端末でのコネクタの利用実態をサーバで把握することができ、各機能利用時に発生する課金処理や認証処理をサーバ側で一括して行なうことが可能となる。

【0223】

【変形例9】

アプリケーションのユーザインタフェースに沿って入力された値を、コネクタを利用して所定のサービス提供サーバ210にアクセスして処理する際に、実際にサービス提供サーバ210においては、所定の関数であるメソッドが利用されて、入力された値の処理が実行される。そこで、その場合の処理について、第9の変形例として説明する。

【0224】

図63は、第9の変形例にかかる端末6300の構成の具体例を示す図である。

【0225】

図63を参照して、端末6300は、アプリケーションデータ保存領域300とアプリケーションデータ管理部320とを含んで構成されるアプリケーション管理部110を含む。アプリケーションデータ保存領域300には占いアプリケーション6310が保存されているものとする。

【0226】

また端末6300は、コネクタデータ保存領域700とコネクタデータ管理部720とを含んで構成されるコネクタ管理部120を含む。コネクタデータ保存領域700には、コネクタデータ6320が保存されているものとする。さらに、コネクタデータ6320には端末外部の占いWebサービス6340を利用するための情報が記述されているものとする。占いWebサービス6340は占い用のメソッドを提供するサービスであるものとする。

【0227】

図64は、占いアプリケーション6310の詳細の具体例を示す図である。

【0228】

図64を参照して、占いアプリケーション6310は、占いを行なうための占いインターフェース6410と、占いの結果を表示するための占い結果画面6420との、2つの画面を含んで構成される。

【0229】

占いインターフェース6410には、占いWebサービスを利用するために入力するデータを入力するフォーム6411～6414が用意されている。本実施例では、名前を入力するためのフォームであるNAME6411、誕生日を入力するためのフォームであるBIRTHDAY6412、血液型を入力するためのフォームであるBLOOD6413、および出身地を入力するためのフォームであるHOMETOWN6414がそれぞれ用意されており、「占う」ボタン6415が押されると、これらのフォームに入力されたデータがコネクタを介して占いWebサービスに送られる。また、メニューボタン6416が押されると、様々なメニューが選択可能に表示される。

【0230】

占い結果画面6420には、結果を表示するための領域RESULT6421と、占いインターフェースに戻るための「戻る」ボタン6422とが用意されている。

【0231】

図65は、占いアプリケーション6310から占いWebサービス6340を利用する場合の処理を説明する図である。図65においては、占いアプリケーション6310から占いWebサービス6340を利用するため、占いWebサービス6340の提供するメソッド6510が示されている。すなわち、占いWebサービス6340の提供するメソッド6510の名前はHoroscopeであり、メソッドの第1引数の型は文字列で名前がnameであり、メソッドの第2引数の型は整数で名前がbirthdayであり、メソッドの戻り値はビットマップ画像であることが示されている。

【0232】

図65を参照して、占いインターフェース6410において、フォームであるNAME6411に値「鈴木二郎」、BIRTHDAY6412に値「1973年3月13日」、BLOOD6413に値「AB型」、HOMETOWN6414に値「広島県」と入力された状態で「占う」ボタン6415が押下されると、占いWebサービス6340を利用するためのコネクタデータ6320が呼出される。

【0233】

コネクタデータ6320には、占いWebサービス6340の存在場所を示すURLや、そこで提供されるメソッドのAPI（アプリケーション）の仕様等が記述されているものとする。コネクタデータ6320にはさらに、占いインターフェース6410から入力されるデータを占いWebサービス6340の提供する占いメソッド6510に引渡すための変換規則、および占いメソッド6510からの戻り値を占い結果画面6420に表示するための変換規則が記述されている。つまり、本変形例においては、コネクタデータに、当該コネクタが決定する機能に対するデータの受け渡しを定義する入出力定義に替えて、当該コネクタが決定する機能を用いるために利用される関数であるメソッドに対するデータの受け渡しを定義する入出力定義が含まれる。

【0234】

具体的なコネクタデータ6320に記述されている、占いメソッド6510を利用するための占いインターフェース6410から入力されるデータを変換する変換規則は以下のとおりである。

規則1) フォームNAME6411に入力された値を文字列に変換して、名前がHoroscopeであるメソッド6510の第1引数とする。

規則2) フォームBIRTHDAY6412に入力された値を整数に変換して、名前がHoroscopeであるメソッド6510の第2引数とする。

規則3) 名前がHoroscopeであるメソッド6510からの戻り値のビットマップ画像をJpeg画像に変換して、領域RESULT6421に表示する。

【0235】

上述の規則1により、コネクタデータ管理部720に含まれる図63においては示されていない外部機能管理部においては、占いインターフェース6410のフォームNAME6411に入力された値「鈴木二郎」は文字列に変換され、占いWebサービス6340のHoroscopeメソッドの第1引数として、図63においては示されていない外部機能呼出部において呼出された占いWebサービス6340に渡される。また、規則2により、占いインターフェース6410のフォームBIRTHDAY6412に入力された値「1973年3月13日」は整数に変換され、占いWebサービス6340のHoroscopeメソッドの第2引数として占いWebサービス6340に渡される。また、規則3により、占いWebサービス6340のHoroscopeメソッドからの戻り値であるビットマップ画像はJpeg画像に変換され、占い結果画面6420の領域RESULT6421に表示される。

【0236】

上記変換規則1～3をXMLにより記述した場合のコネクタデータ6320の一部の具体例を図66に示す。

【0237】

図66を参照して、記述<convert method="Horoscope">において、本記述が、Horoscopeメソッドに対する変換規則であることが宣言される。

【0238】

また、記述<paramConv fromApp="NAME" fromType="String" toService="name" toType="String" />において、アプリケーションから名前NAMEで文字列(String)として渡されるデータを文字列(String)に変換して、サービス側の名前nameの引数に渡すことが宣言される。なお、この場合はどちらも文字列なので、実際の変換は不要となっている。

【0239】

また、記述<paramConv fromApp="BIRTHDAY" fromType="String" toService="birthday" toType="int" />において、アプリケーションから名前BIRTHDAYで文字列(

String) として渡されるデータを整数 (int) に変換して、サービス側の名前 birthday の引数に渡すことが宣言される。

【0240】

また、記述 <returnConv fromService="ReturnValue" fromType="bitmap" toApp="RESULT" toType="jpeg" />において、サービス側からの戻り値であるビットマップ画像 (bitmap) をJpeg画像 (jpeg) に変換して、アプリケーション側の領域 RESULTにおいて表示することが宣言される。

【0241】

また、記述 </convert>において、Horoscopeメソッドに対する変換規則の定義が終了することが宣言される。

【0242】

図66に示される記述にしたがってコネクタデータ6320において占いインターフェース6410から入力されるデータが変換されて占いメソッド6510が利用されると、占いWebサービス6340からの戻り値であるビットマップ画像はJpeg画像に変換され、図65に示されるように占い結果画面6420の領域RESULT6421に表示される。

【0243】

このように、コネクタデータに、当該コネクタが決定する機能に対するデータの受け渡しを定義する入出力定義に替えて、当該コネクタが決定する機能を用いるために利用される関数であるメソッドに対するデータの受け渡しを定義する入出力定義が含まれることで、アプリケーションにおいて入力される値をメソッドの引数に応じた値にする必要がない。そのため、アプリケーションのインターフェースを、利用されるメソッドを考慮して作成する必要がなく、アプリケーションの作成が容易になる。

【0244】

さらに、このようなコネクタデータの構成とすることで、1つのアプリケーションから複数の異なるWebサービスを利用することも可能になる。

【0245】

1つのアプリケーションから複数の異なるWebサービスを利用する場合の、端末6300の構成の具体例を図67に示す。

【0246】

図67を参照して、この場合も、図63に示された場合と同様に、端末6300は、アプリケーションデータ保存領域300とアプリケーションデータ管理部320とを含んで構成されるアプリケーション管理部110を含む。アプリケーションデータ保存領域300には占いアプリケーション6310が保存されているものとする。

【0247】

また端末6300は、コネクタデータ保存領域700とコネクタデータ管理部720とを含んで構成されるコネクタ管理部120を含む。コネクタデータ保存領域700には、コネクタデータ1(6320)とコネクタデータ2(6330)とが保存されているものとする。コネクタデータ1(6320)には端末外部の占いWebサービス1(6340)を利用するための情報が記述され、コネクタデータ2(6330)には端末外部の占いWebサービス2(6350)を利用するための情報が記述されているものとする。

【0248】

占いアプリケーション1(6320)から占いWebサービス1(6340)を利用する場合の処理については、先に図65を用いて説明されている。同様に、占いアプリケーション2(6330)から占いWebサービス2(6350)を利用する場合の処理を、図68を用いて説明する。図68においては、占いアプリケーション2(6330)から占いWebサービス2(6350)を利用するため、Webサービス2(6350)の提供するメソッド6610が示されている。すなわち、Webサービス2(6350)の提供するメソッドの名前はHomeFortuneであり、メソッドの第1引数に型は文

字列で名前が name であり、メソッドの第 2 引数の型は文字列で名前が city であり、メソッドの戻り値は HTML 文書であることが示されている。

【0249】

図 68 を参照して、占いインタフェース 6410において、フォームである NAME 6411 に値「鈴木二郎」、BIRTHDAY 6412 に値「1973年3月13日」、BLOOD 6413 に値「AB型」、HOMETOWN 6414 に値「広島県」と入力された状態で「占う」ボタン 6415 が押下されると、占い Web サービス 2 (6350) を利用するためのコネクタデータ 2 (6330) が呼出される。

【0250】

コネクタデータ 2 (6330) にも、コネクタデータ 1 (6320) と同様に、占い Web サービス 2 (6350) の存在場所を示す URL や、そこで提供されるメソッドの API の仕様等が記述されているものとする。コネクタデータ 2 (6330) にはさらに、占いインタフェース 6410 から入力されるデータを占い Web サービス 2 (6350) の提供する占いメソッド 6610 に引渡すための変換規則、および占いメソッド 6610 からの戻り値を占い結果画面 6420 に表示するための変換規則が記述されている。

【0251】

具体的なコネクタデータ 2 (6330) に記述されている、占いメソッド 6610 を利用するための占いインタフェース 6410 から入力されるデータを変換する変換規則は以下のとおりである。

規則 1) フォーム NAME 6411 に入力された値を文字列に変換して、名前が Home Fortune であるメソッド 6610 の第 1 引数とする。

規則 2) フォーム HOMETOWN 6414 に入力された値を文字列に変換して、名前が Home Fortune であるメソッド 6610 の第 2 引数とする。

規則 3) 名前が Home Fortune であるメソッド 6610 からの戻り値の HTML 文書を、そのまま領域 RESULT 6421 に表示する。

【0252】

上述の規則 1 により、コネクタデータ管理部 720 に含まれる図 67 においては示されていない外部機能管理部においては、占いインタフェース 6410 のフォーム NAME 6411 に入力された値「鈴木二郎」は文字列に変換され、占い Web サービス 2 (6350) の Home Fortune メソッドの第 1 引数として、図 67 においては示されていない外部機能呼出部において呼出された占い Web サービス 6340 に渡される。また、規則 2 により、占いインタフェース 6410 のフォーム HOMETOWN 6414 に入力された値「広島県」は文字列に変換され、占い Web サービス 2 (6350) の Home Fortune メソッドの第 2 引数として占い Web サービス 6340 に渡される。また、規則 3 により、占い Web サービス 2 (6350) の Home Fortune メソッドからの戻り値である HTML 文書は、そのまま占い結果画面 6420 の領域 RESULT 6421 に表示される。

【0253】

上記変換規則 1～3 を XML により記述した場合のコネクタデータ 2 (6330) の一部の具体例を図 69 に示す。

【0254】

図 69 を参照して、記述 <convert method="Home Fortune" >において、本記述が、Home Fortune メソッドに対する変換規則であることが宣言される。

【0255】

また、記述 <paramConv fromApp="NAME" fromType="String" toService="name" toType="String" />において、アプリケーションから名前 NAME で文字列 (String) として渡されるデータを文字列 (String) に変換して、サービス側の名前 name の引数に渡すことが宣言される。なお、この場合はどちらも文字列なので、実際の変換は不要

となっている。

【0256】

また、記述<paramConv fromApp="HOMETOWN" fromType="String" toService="city" toType="String" />において、アプリケーションから名前HOMETOWNで文字列(String)として渡されるデータを文字列(String)に変換して、サービス側の名前cityの引数に渡すことが宣言される。なお、この場合にもどちらも文字列なので、実際の変換は不要となっている。

【0257】

また、記述<returnConv fromService="ReturnValue" fromType="html" toApp="RESULT" toType="HTML" />において、サービス側からの戻り値であるHTML文書(html)をHTML文書(html)に変換して、アプリケーション側の領域RESULTにおいて表示することが宣言される。なお、この場合にもどちらもHTML文書なので、実際の変換は不要となっている。

【0258】

また、記述</convert>において、HomeFortuneメソッドに対する変換規則の定義が終了することを宣言される。

【0259】

図69に示される記述にしたがって、コネクタデータ2(6330)占いインタフェース6410から入力されるデータが変換されて占いメソッド6610が利用されると、占いWebサービス2(6350)からの戻り値であるHTML文章は、図68に示されるように占い結果画面6420の領域RESULT6421に表示される。

【0260】

このように、1つのアプリケーション6310から2つのWebサービス6340, 6350が利用可能な場合、図70および図71に示されるように、接続先であるWebサービス6340, 6350を切替える操作が行なわれる。図70、図71は、アプリケーション6310から利用する占いWebサービス6340, 6350の接続先を変更する方法の具体例を説明する図であって、図70は、2つの占いWebサービス6340, 6350が利用可能な占いアプリケーション6310の、占いインタフェース6410の表示の具体例を示す図である。具体的には、図70に示される占いインタフェース6410の表示は、図64に示される占いインタフェース6410においてメニューボタン6416が押されたときの表示の具体例を示しており、メニューリスト6920が表示されている。

【0261】

図70に示される状態において、さらに、メニューリスト6920に表示された選択項目である接続先変更6930が選択されると、占いインタフェース6410の表示は図71に示される表示に遷移する。すなわち、図70に示される占いインタフェース6410の表示において接続先変更6930が選択されると、図71に示されるように、選択可能な接続先として選択項目がリスト表示された占いインタフェース6410が表示される。本変形例においては、選択項目として占いサービス1(星占い)7010と、占いサービス2(出身地占い)7020との2つがリスト表示される。

【0262】

図71に示される占いインタフェース6410の表示において、接続先を占いサービス1(7010)と占いサービス2(7020)との中から選択して変更ボタン7030を押下することで、接続先が変更される。つまり、図71に示される占いインタフェース6410の表示において接続先が選択されることで、占いアプリケーション6310において占いWebサービス1(6340)または占いWebサービス2(6350)を利用するため呼出すコネクタデータとしてコネクタデータ1(6320)またはコネクタデータ1(6330)が決定され、コネクタデータ管理部720において、決定されたコネク

タデータの記述内容にしたがってここでは図示されていない外部機能管理部の機能が利用されて占いWebサービス1(6340)または占いWebサービス2(6350)の機能が呼出される。

【0263】

このように、第9の変形例においては、端末630においてアプリケーションデータとコネクタデータとが分割されて各々別個に管理されているため、単一のアプリケーションデータから複数の異なるWebサービスを利用することが可能になる。そのため、アプリケーションの形態がWebサービスに関わらず、アプリケーションの作成が容易になる。また、その場合に、いずれのWebサービスを利用する場合であっても、同じユーザインターフェースにおいてデータ入力を行なうことができるため、ユーザの操作が容易になる。

【0264】

さらに、上述のサービス利用端末におけるサービス利用方法やコネクタ提供サーバにおけるコネクタ提供処理を、プログラムとして提供することもできる。このようなプログラムは、フレキシブルディスク、CD-ROM、ROM、RAMおよびメモリカードなどのコンピュータ読取り可能な記録媒体にて記録させて、プログラム製品として提供することもできる。あるいは、コンピュータに内蔵するハードディスクなどの記録媒体にて記録させて、プログラムを提供することもできる。また、ネットワークを介したダウンロードによって、プログラムを提供することもできる。

【0265】

提供されるプログラム製品は、ハードディスクなどのプログラム格納部にインストールされて実行される。なお、プログラム製品は、プログラム自体と、プログラムが記録された記録媒体とを含む。

【0266】

今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【図面の簡単な説明】

【0267】

【図1】本実施の形態におけるサービス提供システムの構成の具体例を示す図である。

【図2】本実施の形態におけるサービス利用端末100のハードウェア構成の具体例を示す図である。

【図3】本実施の形態におけるサービス利用端末100の機能構成の具体例を示す図である。

【図4】アプリケーション管理部110の構成の具体例を示す図である。

【図5】アプリケーションデータ310の具体例を示す図である。

【図6】アプリケーションデータ310のうち、アプリケーション名401、アプリケーション識別子402、有効期限403、および最大利用回数404の具体例を示す図である。

【図7】アプリケーション405の具体例を示す図である。

【図8】サービス利用端末100の、コネクタ管理部120の構成の具体例を示す図である。

【図9】コネクタデータ710の具体例を示す図である。

【図10】コネクタデータ710のうち、コネクタ名801、コネクタ識別子802、有効期限803、および最大利用回数804の具体例を示す図である。

【図11】コネクタデータ710のうち、コネクタ名801、コネクタ識別子802、有効期限803、および最大利用回数804の他の具体例を示す図である。

【図12】入出力定義805aの具体例を示す図である。

【図13】入出力定義805bの具体例を示す図である。

【図14】外部機能管理部130の具体例を示す図である。

【図15】内部機能管理部140の具体例を示す図である。

【図16】簡略化したサービス提供システムの構成の具体例を示す図である。

【図17】サービス利用端末1301においてアプリケーションを実行する際の処理を示すフローチャートである。

【図18】入力画面の具体例を示す図である。

【図19】ステップS109における端末外機能利用処理を示すフローチャートである。

【図20】画面表示の具体例を示す図である。

【図21】画面表示の具体例を示す図である。

【図22】ステップS110における端末内機能利用処理を示すフローチャートである。

【図23】画面表示の具体例を示す図である。

【図24】画面表示の具体例を示す図である。

【図25】画面表示の具体例を示す図である。

【図26】画面表示の具体例を示す図である。

【図27】画面表示の具体例を示す図である。

【図28】本発明にかかるサービス利用端末が実行するサービス利用処理の概要を表わす図である。

【図29】アプリケーション405の他の具体例を示す図である。

【図30】画面表示の具体例を示す図である。

【図31】アプリケーション405のさらに他の具体例を示す図である。

【図32】画面表示の具体例を示す図である。

【図33】コネクタデータ710の他の具体例を示す図である。

【図34】画面表示の具体例を示す図である。

【図35】第4の変形例におけるサービス提供システムの、簡略化した構成の具体例を示す図である。

【図36】アプリケーションデータ310の他の具体例を示す図である。

【図37】コネクタデータ710の他の具体例を示す図である。

【図38】第4の変形例における端末外機能利用処理を示すフローチャートである。

【図39】従来の一般的なWebサービスの構成の具体例を示す図である。

【図40】本実施の形態におけるコネクタ提供サーバ280のハードウェア構成の具体例を示す図である。

【図41】本実施の形態におけるコネクタ提供サーバ280の機能構成の具体例を示す図である。

【図42】コネクタ提供サーバ280の、コネクタ管理部520の構成の具体例を示す図である。

【図43】コネクタデータ710の他の具体例を示す図である。

【図44】コネクタ提供サーバ280におけるコネクタの受信処理を示すフローチャートである。

【図45】サービス提供サーバ情報テーブルの具体例を示す図である。

【図46】アプリケーション405の具体例を示す図である。

【図47】コネクタ提供サーバ1307におけるコネクタ送信処理を示すフローチャートである。

【図48】サービス提供システムの、簡略化した構成の他の具体例を示す図である。

【図49】入出力定義の具体例を示す図である。

【図50】サービス提供システムの、簡略化した構成の他の具体例を示す図である。

【図51】入出力定義の具体例を示す図である。

【図52】第5の変形例のコネクタ提供サーバ1307におけるサービス中継処理を示すフローチャートである。

【図53】第6の変形例にかかるアプリケーション405の具体例を示す図である。

【図54】入力画面の具体例を示す図である。

【図55】情報入力後の入力画面の具体例を示す図である。

【図56】第7の変形例にかかるアプリケーション405の具体例を示す図である。

【図57】表示画面の具体例を示す図である。

【図58】表示画面の具体例を示す図である。

【図59】第7の変形例にかかるアプリケーション405の他の具体例を示す図である。

【図60】表示画面の具体例を示す図である。

【図61】表示画面の具体例を示す図である。

【図62】第8の変形例にかかるコネクタデータの保存方法および管理方法の具体例を示す図である。

【図63】第9の変形例にかかる端末6300の構成の具体例を示す図である。

【図64】占いアプリケーション6310の詳細の具体例を示す図である。

【図65】占いアプリケーション6310から占いWebサービス6340を利用する場合の処理を説明する図である。

【図66】変換規則1～3をXMLにより記述した場合のコネクタデータ6320の一部の具体例を示す図である。

【図67】1つのアプリケーションから複数の異なるWebサービスを利用する場合の、端末6300の構成の具体例を示す図である。

【図68】占いアプリケーション2(6330)から占いWebサービス2(6350)を利用する場合の処理を説明する図である。

【図69】変換規則1～3をXMLにより記述した場合のコネクタデータ2(6330)の一部の具体例を示す図である。

【図70】アプリケーション6310から利用する占いWebサービス6340, 6350の接続先を変更する方法の具体例を説明する図である。

【図71】アプリケーション6310から利用する占いWebサービス6340, 6350の接続先を変更する方法の具体例を説明する図である。

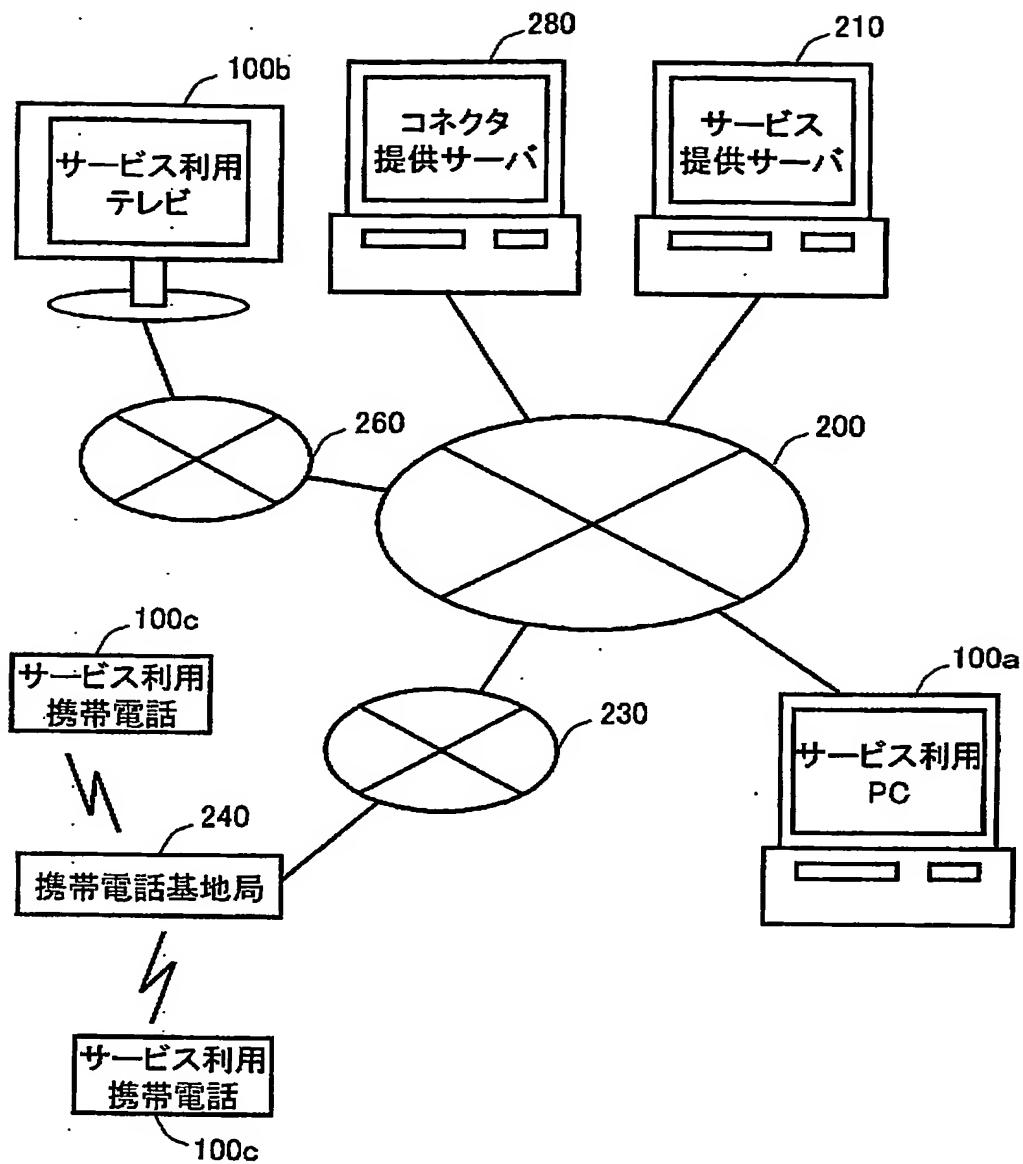
【符号の説明】

【0268】

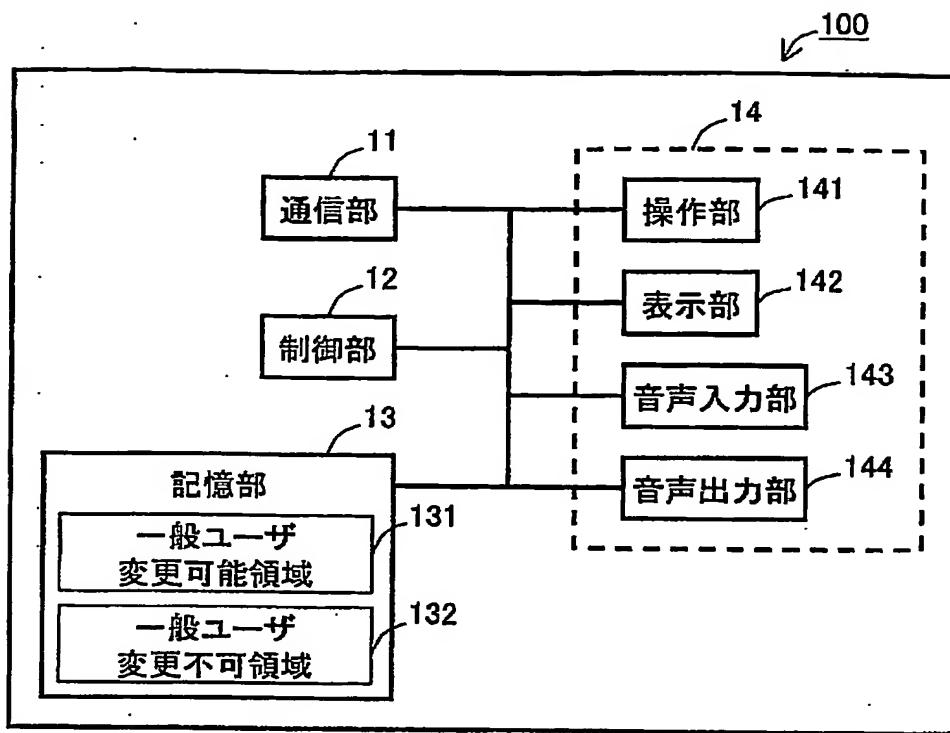
11 サービス利用端末の通信部、12 サービス利用端末の制御部、13 サービス利用端末の記憶部、14 入出力部、21 コネクタ提供サーバの通信部、22 コネクタ提供サーバの制御部、23 コネクタ提供サーバの記憶部、24 コネクタ提供サーバの操作部、25 コネクタ提供サーバの表示部、100, 1301 サービス利用端末、131 一般ユーザ変更可能領域、132 一般ユーザ変更不可領域、141 サービス利用端末の操作部、142 サービス利用端末の表示部、143 音声入力部、144 音声出力部、110 アプリケーション管理部、120 サービス利用端末のコネクタ管理部、130 外部機能管理部、140 内部機能管理部、150 サービス利用端末の通信制御部、160 サービス利用端末の記憶領域、200 ネットワーク、210, 1306 サービス提供サーバ、230 携帯電話網、231 プログラム領域、232 データ領域、240 携帯電話基地局、260 ホームネットワーク、280, 1307 コネクタ提供サーバ、300 アプリケーションデータ保存領域、310 アプリケーションデータ、320 アプリケーションデータ管理部、401 アプリケーション名、402 アプリケーション識別子、403, 803 有効期限、404, 804 最大利用回数、405 アプリケーション、510 端末管理部、520 コネクタ提供サーバのコネクタ管理部、530 コネクタ提供サーバの通信制御部、540 コネクタ提供サーバの記憶領域、700 サービス利用端末のコネクタデータ保存領域、710, 910 コネクタデータ、720 サービス利用端末のコネクタデータ管理部、801 コネクタ名、802 コネクタ識別子、805 入出力定義、900 コネクタ提供サーバのコネクタデータ保存領域、920 コネクタ提供サーバのコネクタデータ管理部、1101

外部機能呼出部、1102 外部機能結果解析部、1201 内部機能呼出部、120
2 内部機能結果解析部、1302 ディスプレイ、1303～1305 機能、130
8 中継サーバ、6310 占いアプリケーション、6320, 6330 コネクタデー
タ、6340, 6350 占いWebサービス、6410 占いインターフェース、641
1～6414 フォーム、6415, 6416, 6422, 7030 ボタン、6420
占い結果画面、6421 領域、6510, 6610 メソッド、6920 メニュー
リスト、6930, 7010, 7020 選択項目。

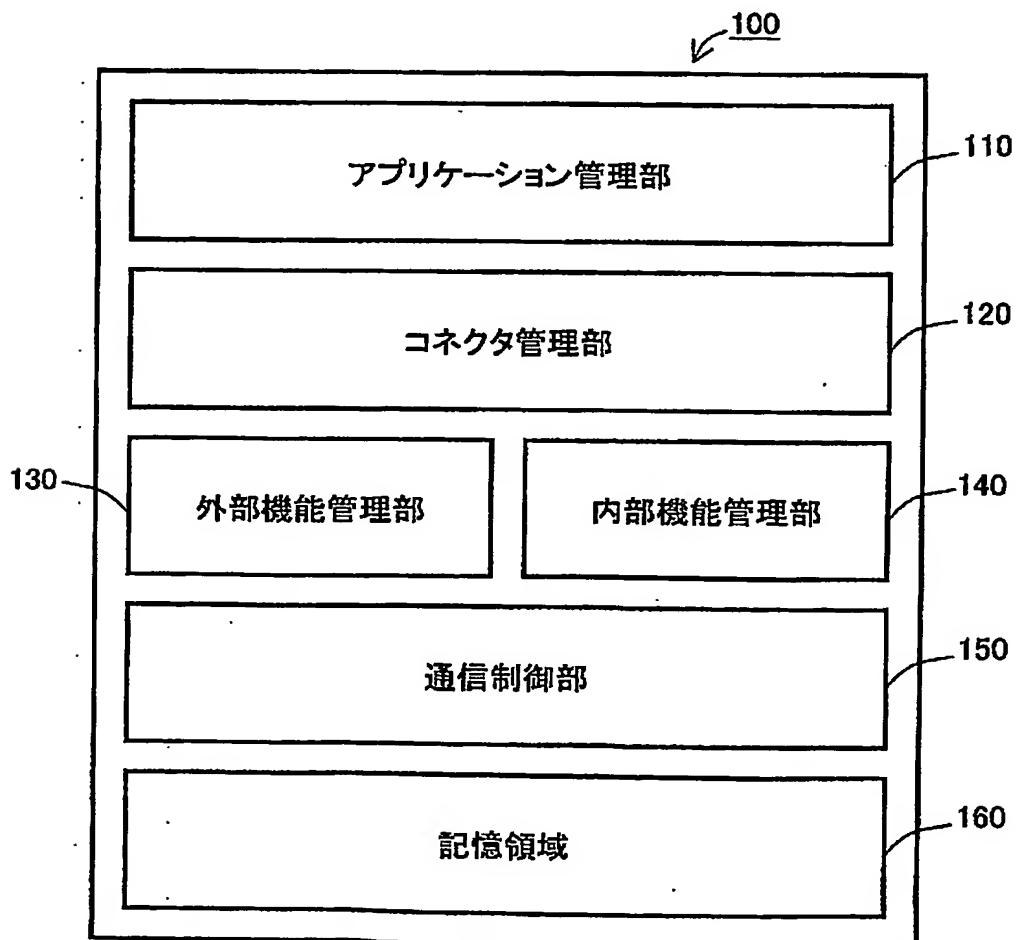
【書類名】図面
【図1】



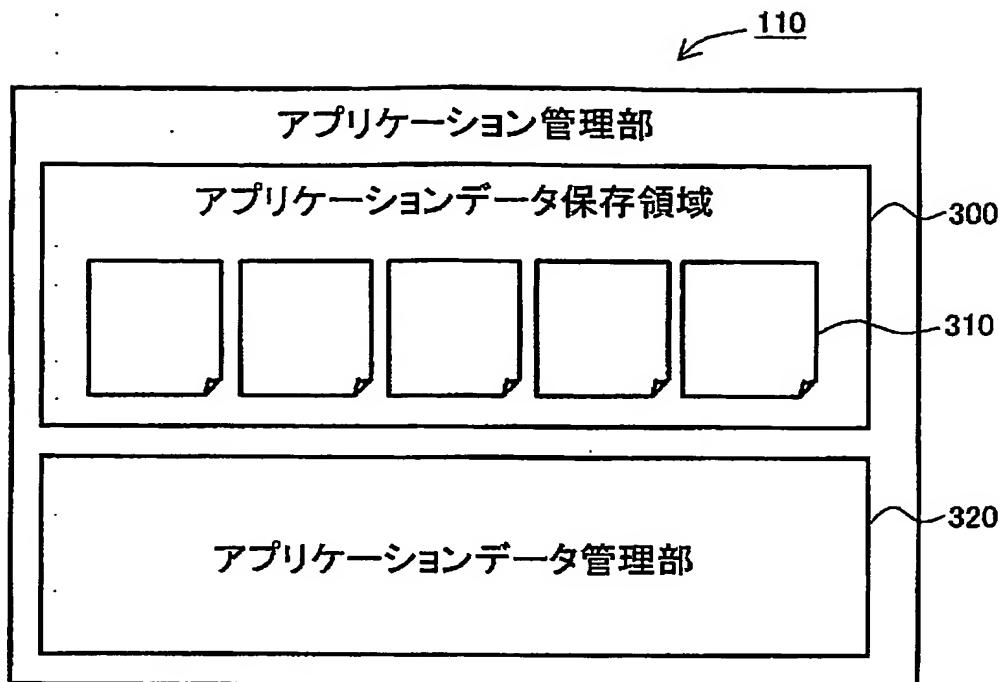
【図2】



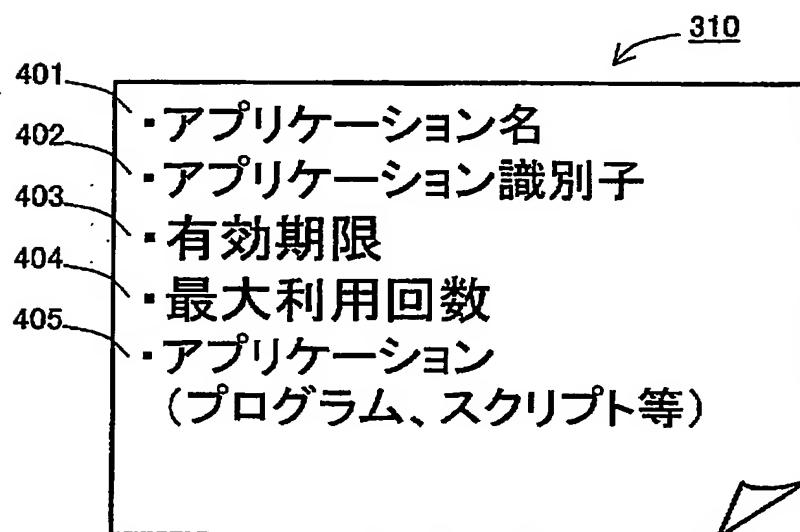
【図3】



【図4】



【図5】



【図 6】

310

```
<application>
  <name>
    401 { Multimedia Message
    </name>
  <id>
    <vendor>
      402 { AAA Corp.
      <vendor/>
      <uniqueNumber>
        HG3490S2345
      </uniqueNumber>
    </id>
    <expireDate>
      403 { 2004/4/1
      </expireDate>
      <expireNumber>
        404 { 100
        </expireNumber>
      </application>
```

【図7】

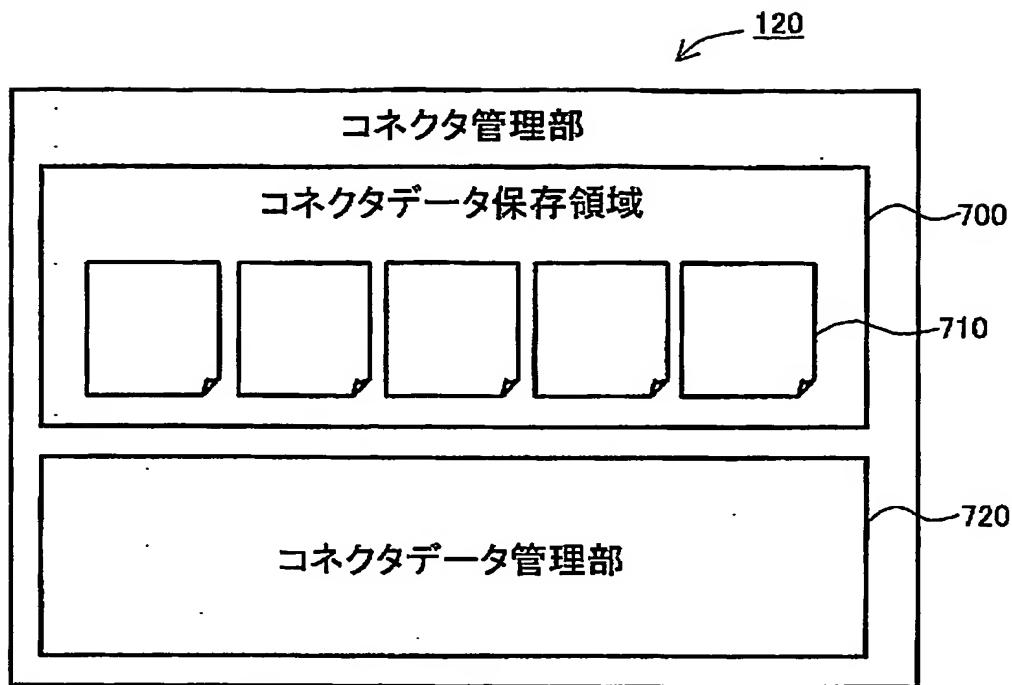
405

```

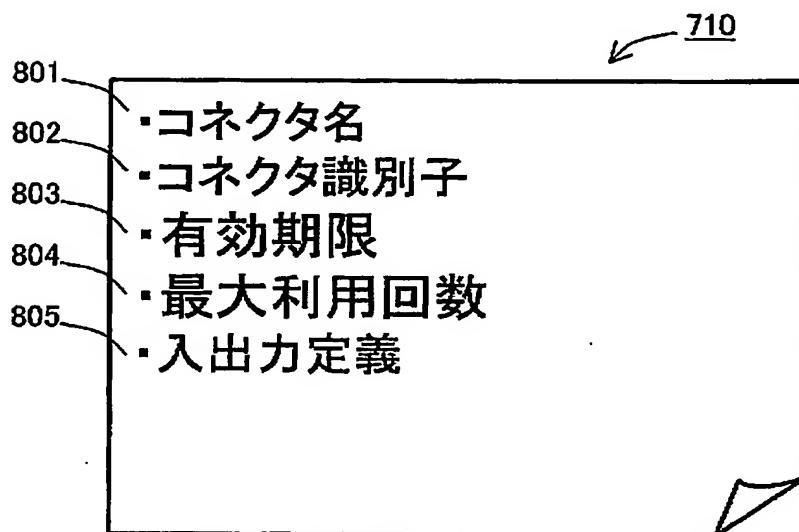
<smil>
  <head>
    <layout>
      <root-layout width="200" height="200"/>
      <region id="Image" width="150" height="100"
        left="25" top="10" />
      <region id="Text" width="180" height="30"
        left="10" top="120" />
      <button id="TextArt" width="40" height="30"
        left="10" top="160" />
      <button id="Camera" width="40" height="30"
        left="60" top="160" />
      <button id="File" width="30" height="30"
        left="110" top="160" />
      <button id="Send" width="40" height="30"
        left="150" top="160" />
    </layout>
    <action id="TextArt" >
      <setMedia func="name:TextArt" src="Text"
        dest="Text" scope="Terminal" />
    </action>
    <action id="Camera" >
      <setMedia func="name:Camera" src=""
        dest="Image" scope="Terminal" />
    </action>
    <action id="File" >
      <setMedia func="name:File" src=""
        dest="Image" scope="Terminal" />
    </action>
    <action id="Send" >
      <setMedia func="name:Send" src=""
        dest="" scope="Terminal" />
    </action>
  </head>
  <body>
    <par>
      <event value="TextArt" action="TextArt" />
      <event value="Camera" action="Camera" />
      <event value="File" action="File" />
      <event value="Send" action="Send" />
    </par>
  </body>
</smil>

```

【図8】



【図9】



【図10】

710 ↗

```
<connector>
  <name>
    TextArt
  </name>
  <id>
    <vendor>
      AAA Corp.
    <vendor/>
    <vendorLimit>
      AAA Coop.
    <vendorLimit/>
    <uniqueNumber>
      LS4369P3550
    </uniqueNumber>
  </id>
  <expireDate>
    2004/4/1
  </expireDate>
  <expireNumber>
    100
  </expireNumber>
</connector>
```

801 {

802 {

803 {

804 {

【図 1 1】

710

```

<connector>
  <name>
    Camera
  </name>
  <id>
    <vendor>
      AAA Corp.
    </vendor>
    <vendorLimit>
      None
    </vendorLimit>
    <uniqueNumber>
      RE6589P8112
    </uniqueNumber>
  </id>
  <expireDate>
    None
  </expireDate>
  <expireNumber>
    None
  </expireNumber>
</connector>

```

【図 1 2】

805a

```

<入出力定義>
  <型定義 名前="TextArtリクエスト" 形式="文字列" />
  <型定義 名前="TextArtレスポンス" 形式="Jpeg画像" />

  <機能定義 名前="TextArt">
    <入力値 型="TextArtリクエスト" />
    <出力値 型="TextArtレスポンス" />
  </機能定義>

  <サービス定義 名前="TextArtサービス" />
  <アドレス URL="http://example.url/TextArt" />
  <利用機能 名前="TextArt" />
</サービス定義>
</入出力定義>

```

【図13】

805b

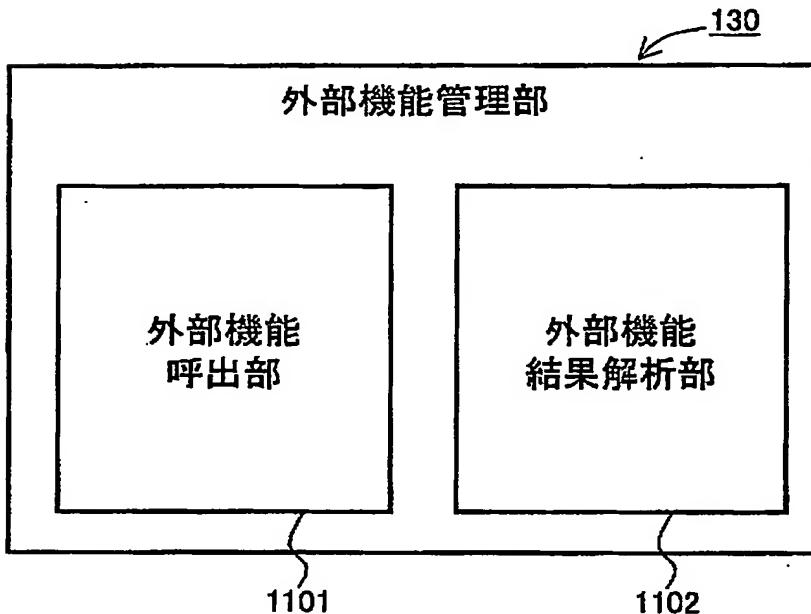
```

<入出力定義>
2601 └─<型定義 名前="Cameraレスポンス" 形式="Jpeg画像" />

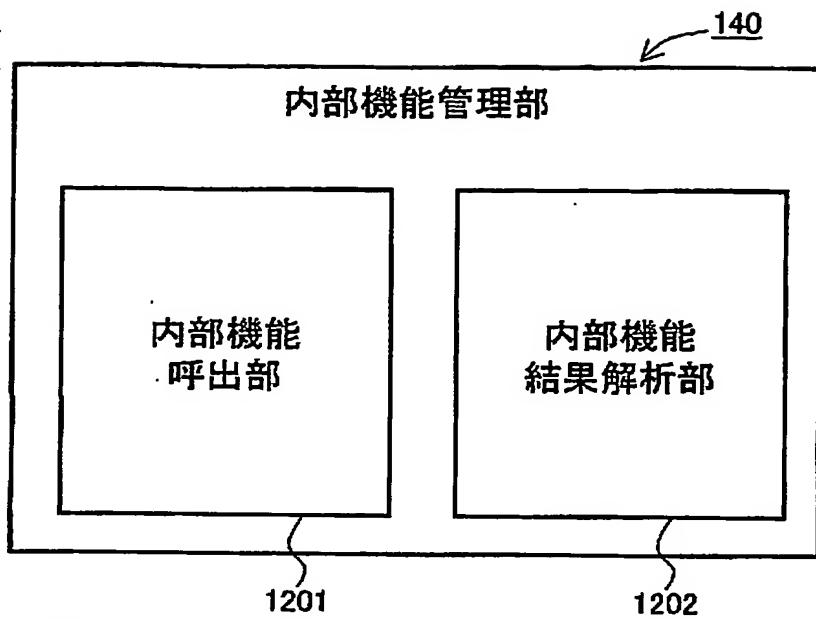
<機能定義 名前="Camera">
2602   <入力値 型="None" />
   <出力値 型="Cameraレスポンス" />
   </機能定義>
   </入出力定義>

```

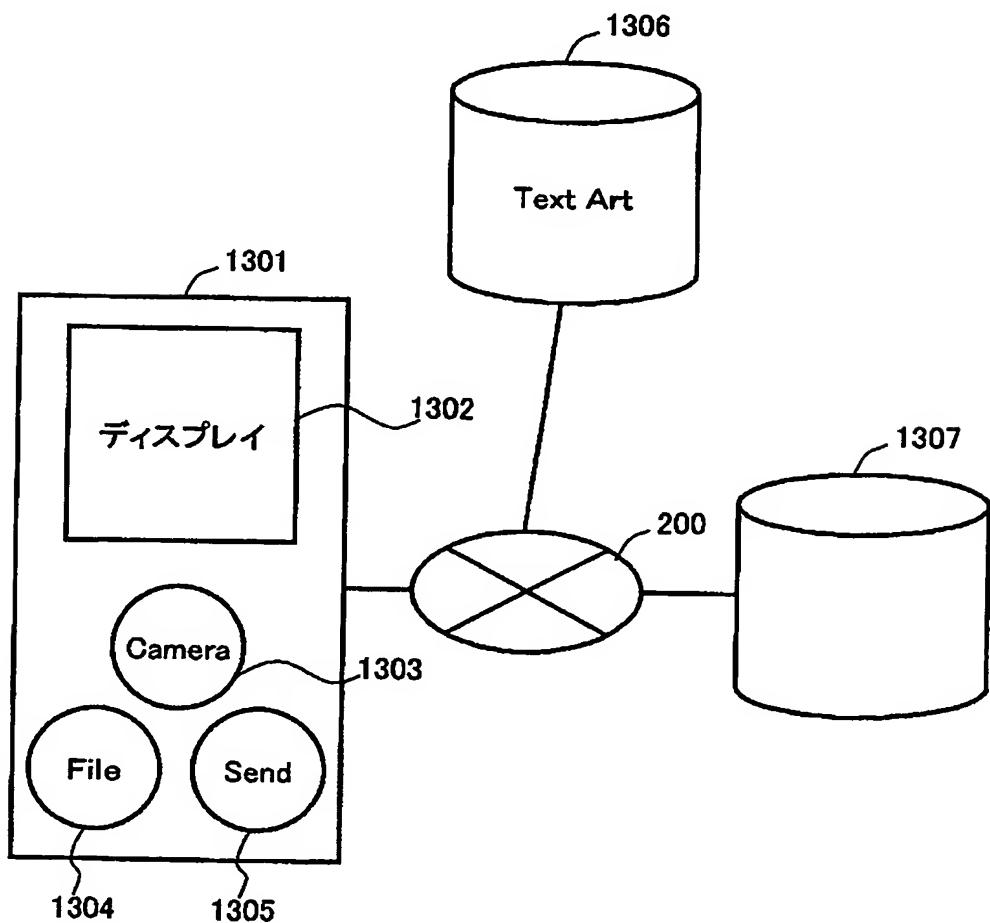
【図14】



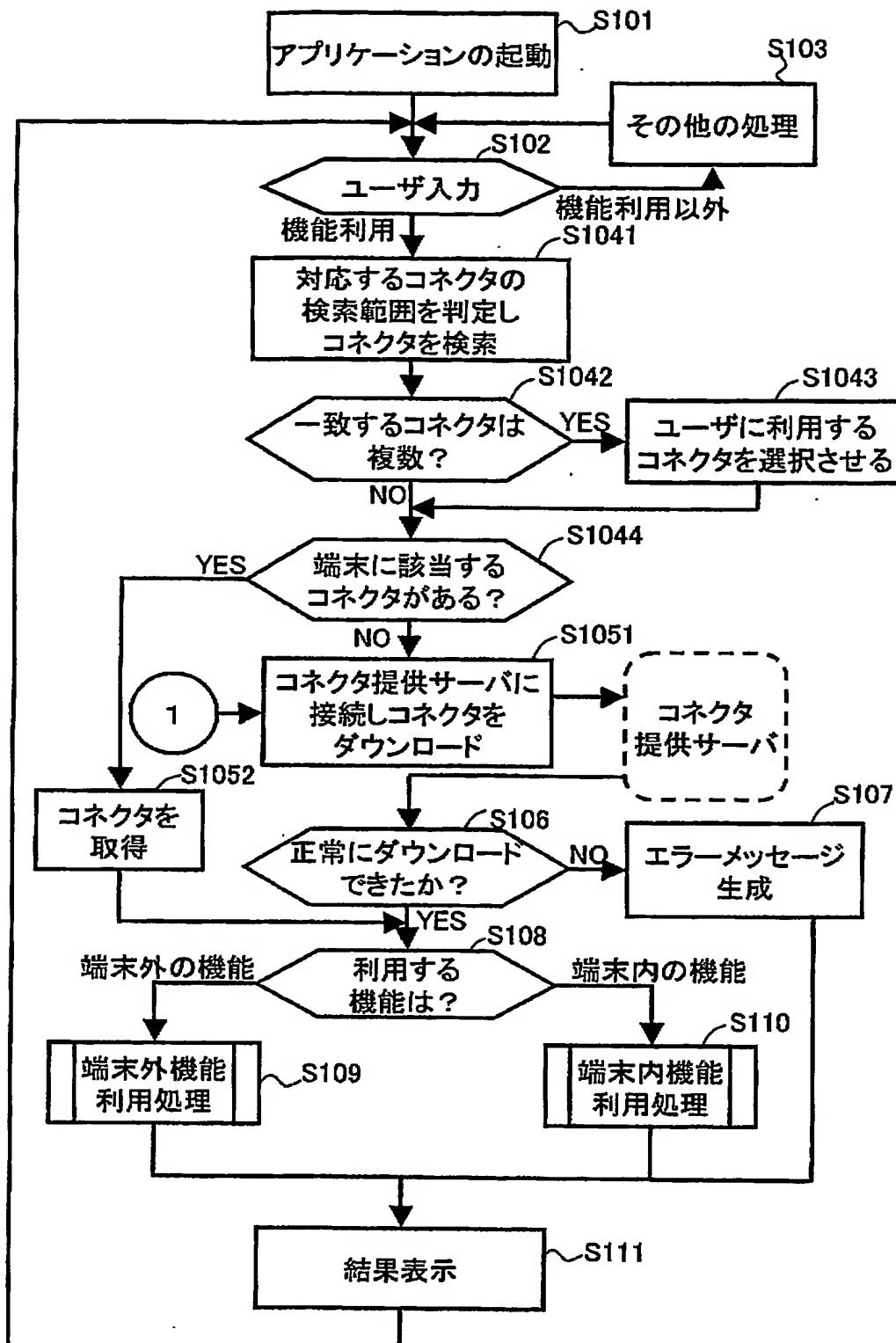
【図15】



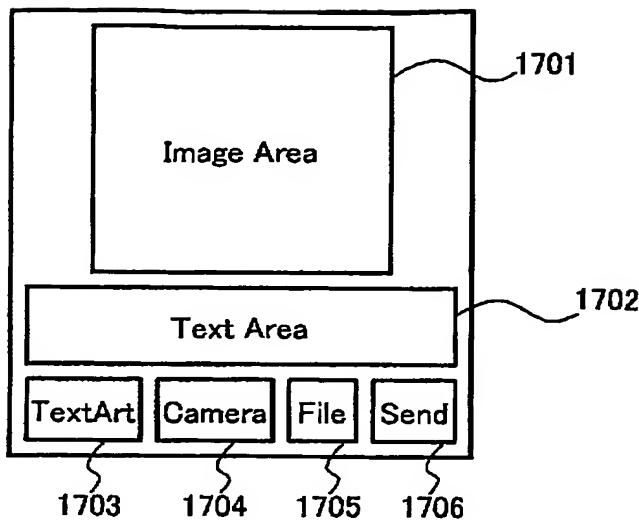
【図16】



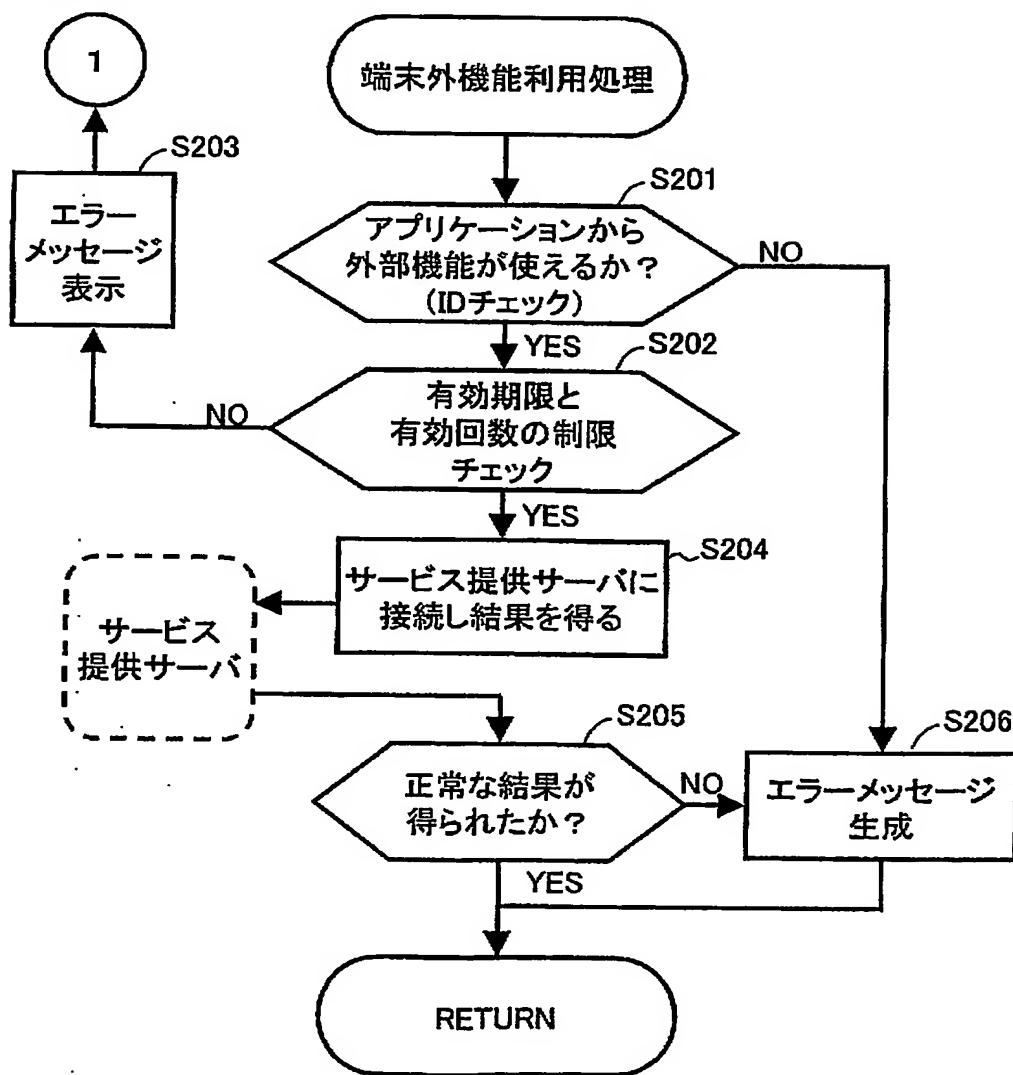
【図17】



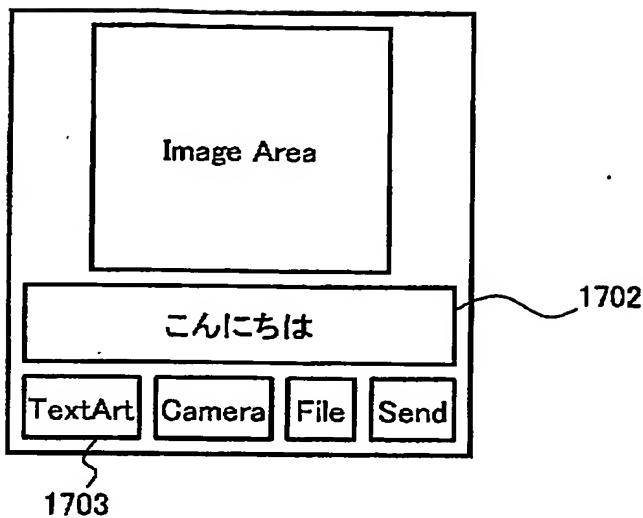
【図18】



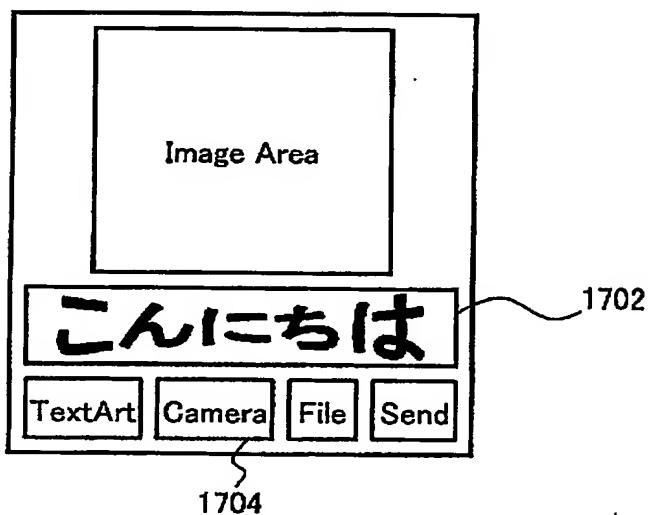
【図19】



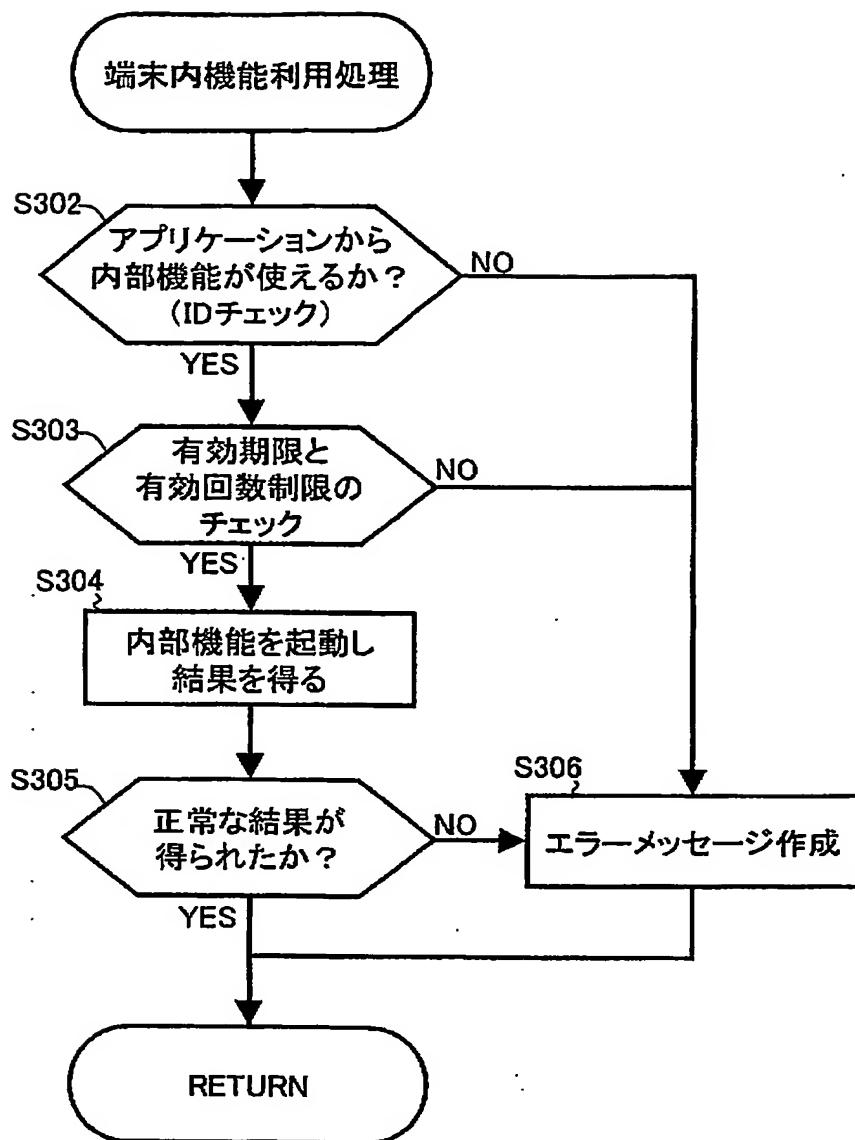
【図20】



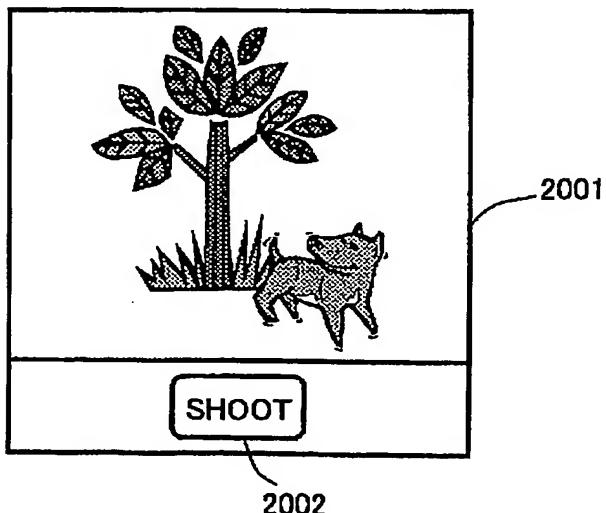
【図21】



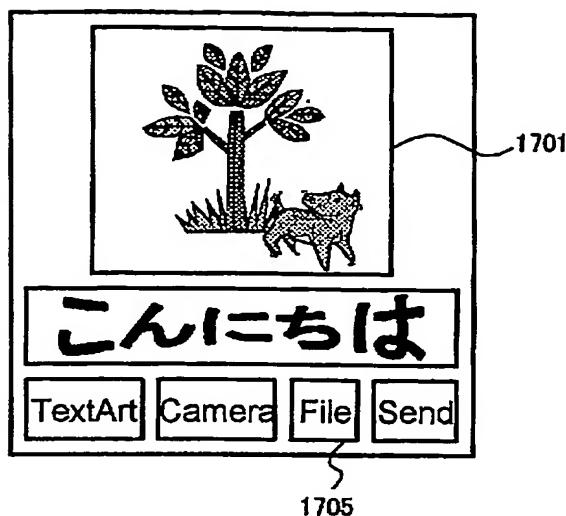
【図22】



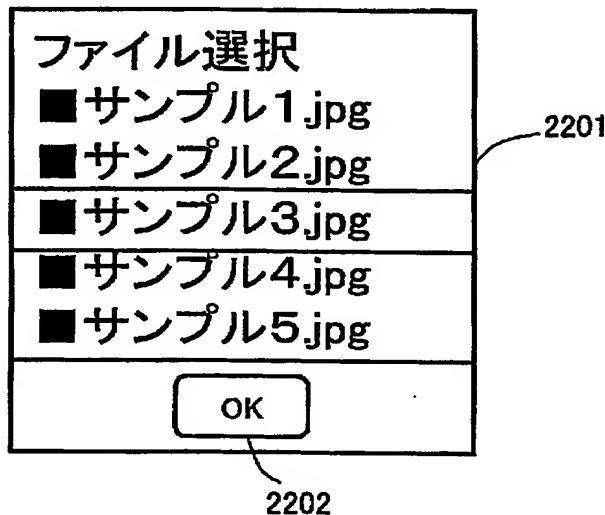
【図23】



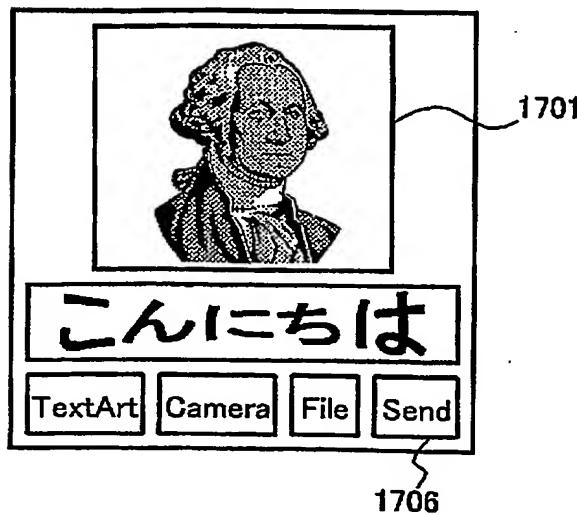
【図24】



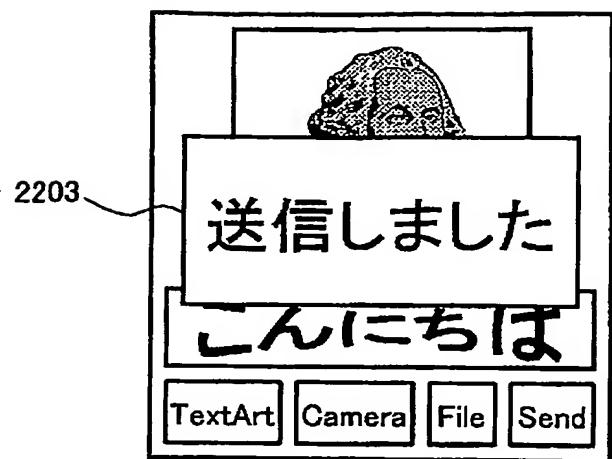
【図25】



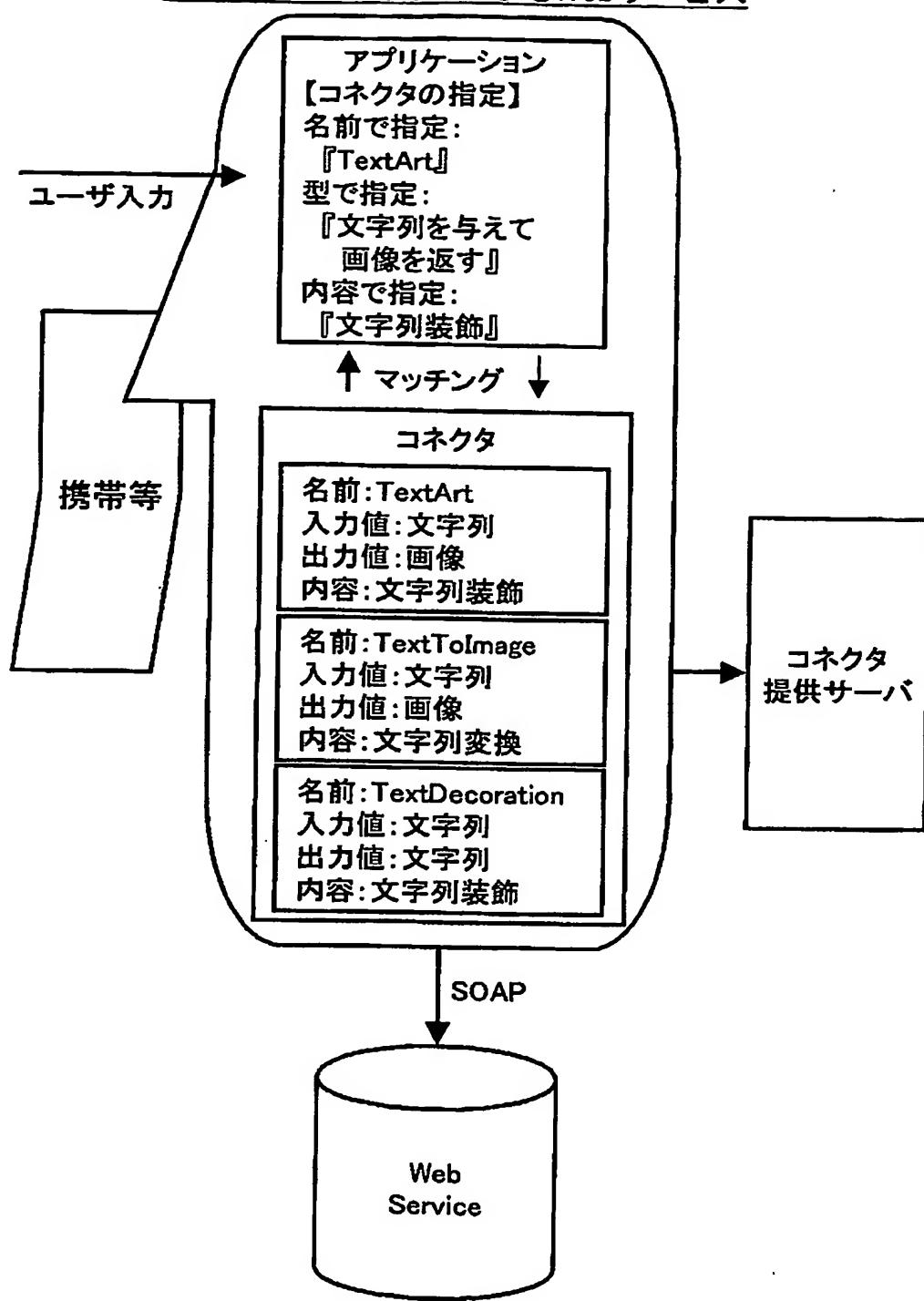
【図26】



【図27】



【図28】

アプリケーションとコネクタによるWebサービス

【図 29】

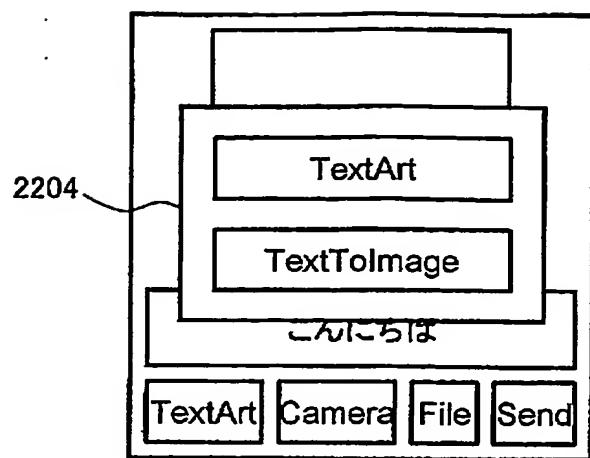
405

```

<smil>
  <head>
    <layout>
      <root-layout width="200" height="200"/>
      <region id="Image" width="150" height="100"
        left="25" top="10" />
      <region id="Text" width="180" height="30"
        left="10" top="120" />
      <button id="TextArt" width="40" height="30"
        left="10" top="160" />
      <button id="Camera" width="40" height="30"
        left="60" top="160" />
      <button id="File" width="30" height="30"
        left="110" top="160" />
      <button id="Send" width="40" height="30"
        left="150" top="160" />
    </layout>
  601  </head>
  6021 <action id="TextArt" >
    <setMedia func="input:文字列,output:Jpeg画像"
      src="Text" dest="Image" scope="Terminal+Server" />
  </action>
  602  <action id="Camera" >
    <setMedia func="name:Camera" src=""
      dest="Image" scope="Terminal" />
  </action>
  <action id="File" >
    <setMedia func="name:File" src=""
      dest="Image" scope="Terminal" />
  </action>
  <action id="Send" >
    <setMedia func="name:Send" src=""
      dest="" scope="Terminal" />
  </action>
  </head>
  <body>
    <par>
      603   <event value="TextArt" action="TextArt" />
      <event value="Camera" action="Camera" />
      <event value="File" action="File" />
      <event value="Send" action="Send" />
    </par>
  </body>
</smil>

```

【図30】



【図31】

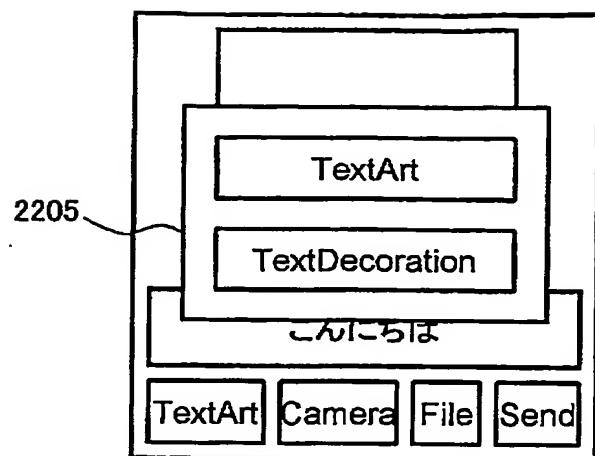
405

```

<smil>
  <head>
    <layout>
      <root-layout width="200" height="200"/>
      <region id="Image" width="150" height="100"
        left="25" top="10" />
      <region id="Text" width="180" height="30"
        left="10" top="120" />
      <button id="TextArt" width="40" height="30"
        left="10" top="160" />
      <button id="Camera" width="40" height="30"
        left="60" top="160" />
      <button id="File" width="30" height="30"
        left="110" top="160" />
      <button id="Send" width="40" height="30"
        left="150" top="160" />
    </layout>
  <601>
    <action id="TextArt" >
      <setMedia func="desc:文字列装飾" src="Text"
        dest="Text" scope="Terminal+Server" />
    </action>
  <602>
    <action id="Camera" >
      <setMedia func="name:Camera" src=""
        dest="Image" scope="Terminal" />
    </action>
    <action id="File" >
      <setMedia func="name:File" src=""
        dest="Image" scope="Terminal" />
    </action>
    <action id="Send" >
      <setMedia func="name:Send" src=""
        dest="" scope="Terminal" />
    </action>
  </602>
  </head>
  <body>
    <603>
      <par>
        <event value="TextArt" action="TextArt" />
        <event value="Camera" action="Camera" />
        <event value="File" action="File" />
        <event value="Send" action="Send" />
      </par>
    </body>
  </smil>

```

【図32】



【図 3 3】

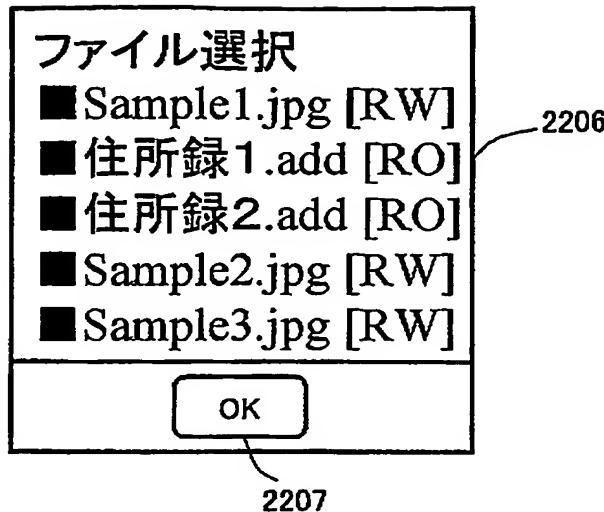
710
↙

```
<connector>
  <name>
    801 File
  </name>
  <id>
    <vendor>
      AAA Corp.
    <vendor/>
    <vendorLimit>
      802 None
    <vendorLimit/>
    <uniqueNumber>
      GH3240LS1234
    </uniqueNumber>
  </id>
  <expireDate>
    803 None
  </expireDate>
  <expireNumber>
    804 None
  </expireNumber>

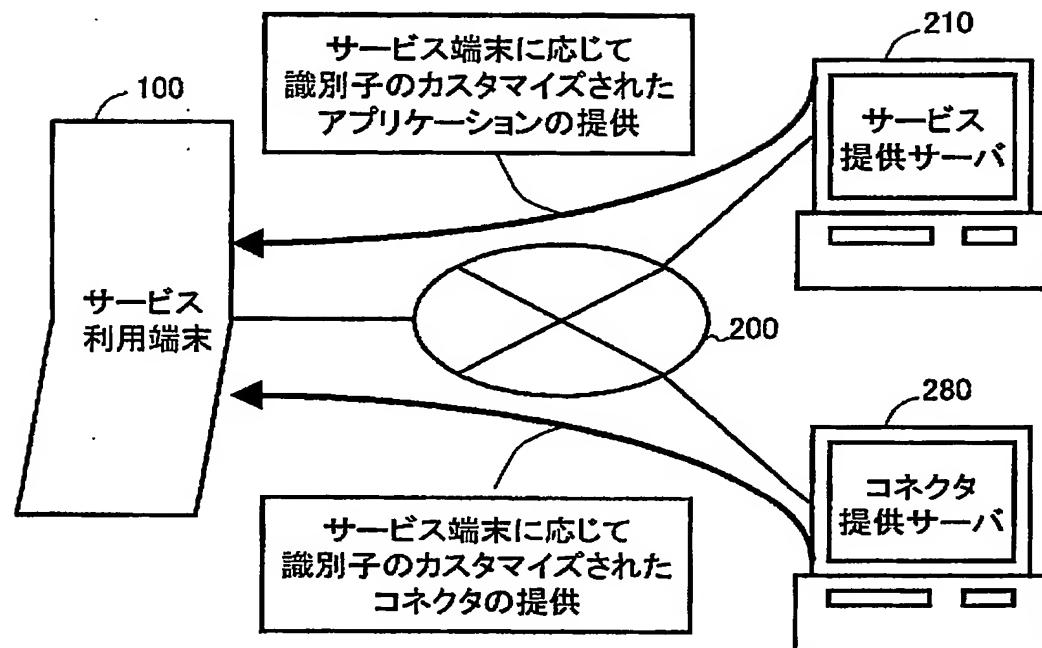
  <accessControl>
    <file type="システム" prop="Hidden"
          limit="ReadOnly"/>
    <file type="住所録" prop="Visible"
          limit="ReadOnly" />
    <file type="一般" prop="Visible"
          limit="ReadWrite" />
  </accessControl>

</connector>
```

【図34】



【図35】



【図36】

310

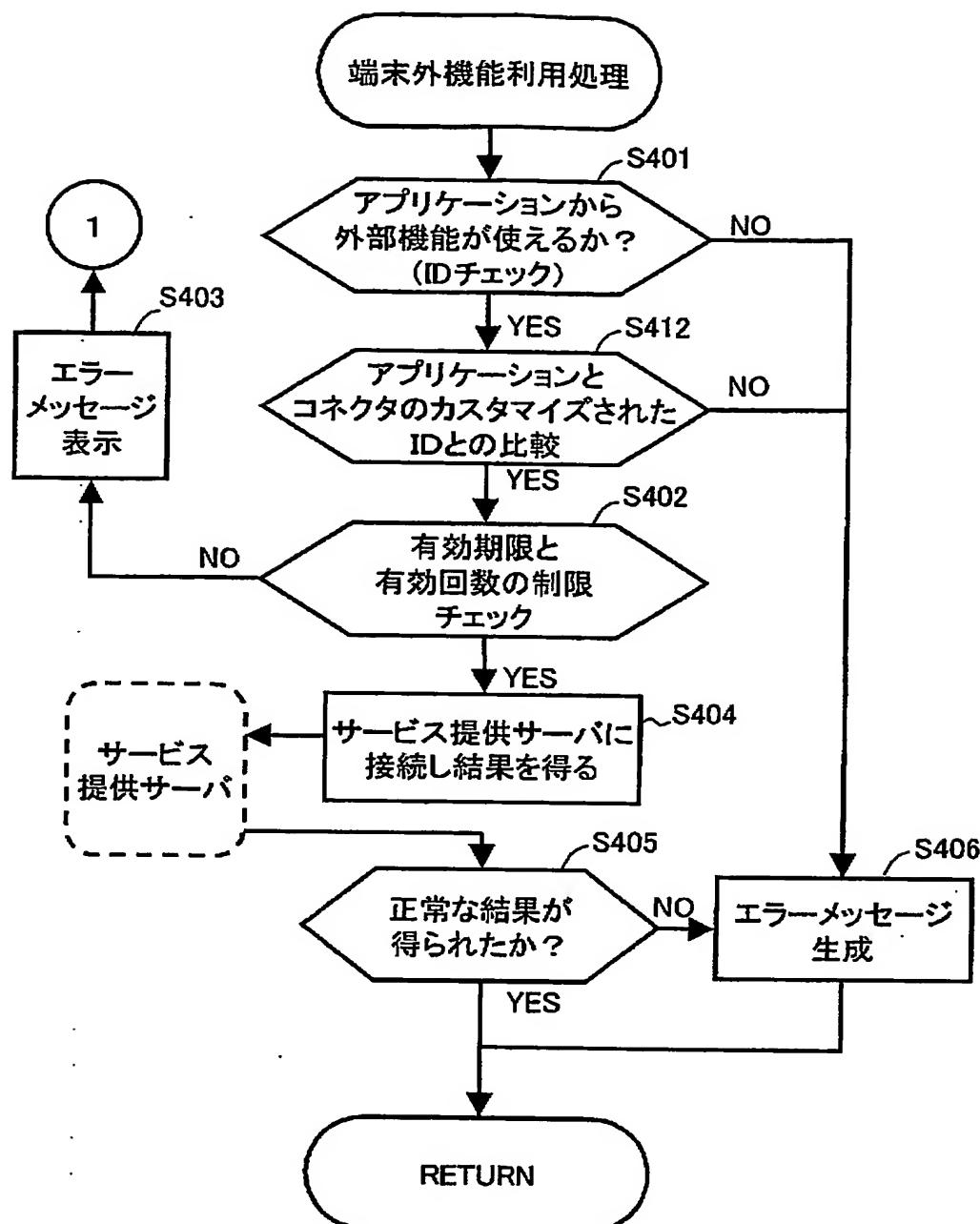
```
<application>
  <name>
    401 { Multimedia Message
  </name>
  <id>
    <vendor>
      402 { AAA Corp.
    <vendor/>
    <uniqueNumber>
      MH0313-0987-0001-A
    </uniqueNumber>
    </id>
    <expireDate>
      403 { 2004/4/1
    </expireDate>
    <expireNumber>
      404 { 100
    </expireNumber>
  </application>
```

【図 37】

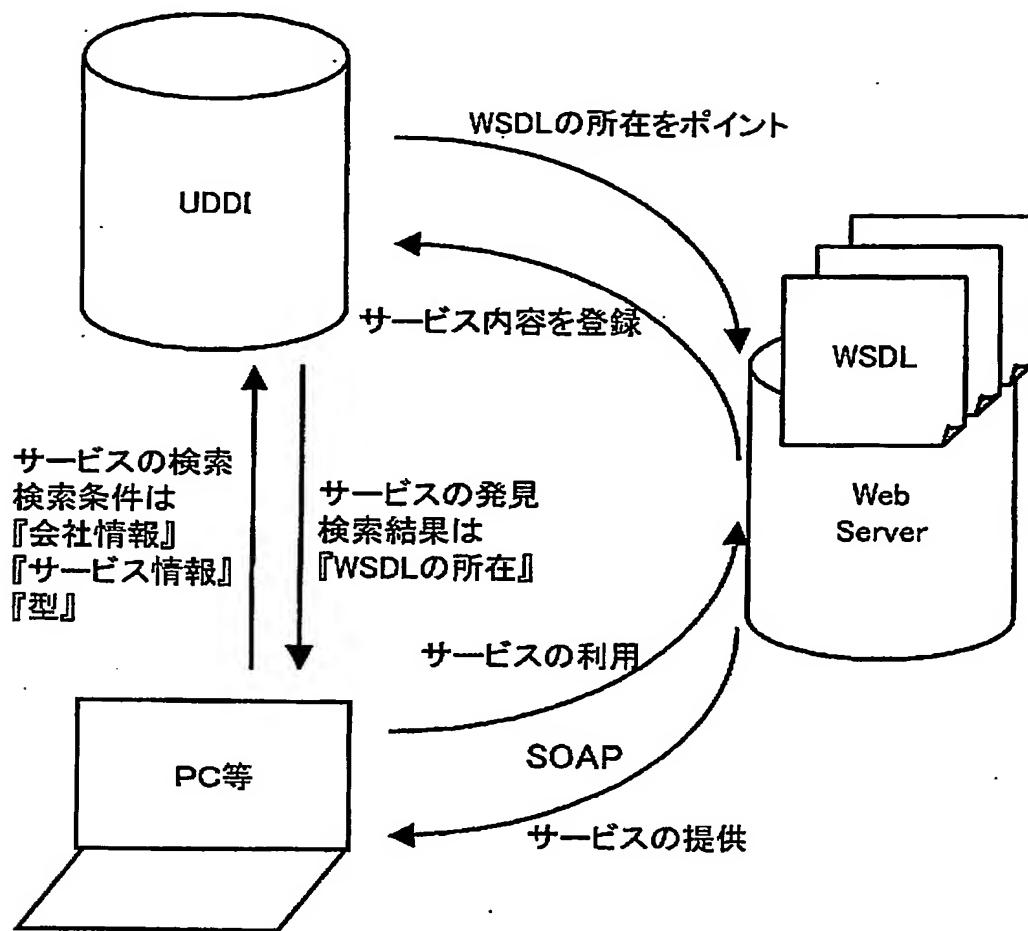
710

```
<connector>
 801  { <name>
        TextArt
        </name>
        <id>
          <vendor>
            AAA Corp.
          <vendor/>
          <vendorLimit>
            AAA Coop.
          <vendorLimit/>
          <uniqueNumber>
            MH0313-1928-9843-C
          </uniqueNumber>
        </id>
        <expireDate>
          2004/4/1
        </expireDate>
        <expireNumber>
          100
        </expireNumber>
      }
 802  { </connector>
```

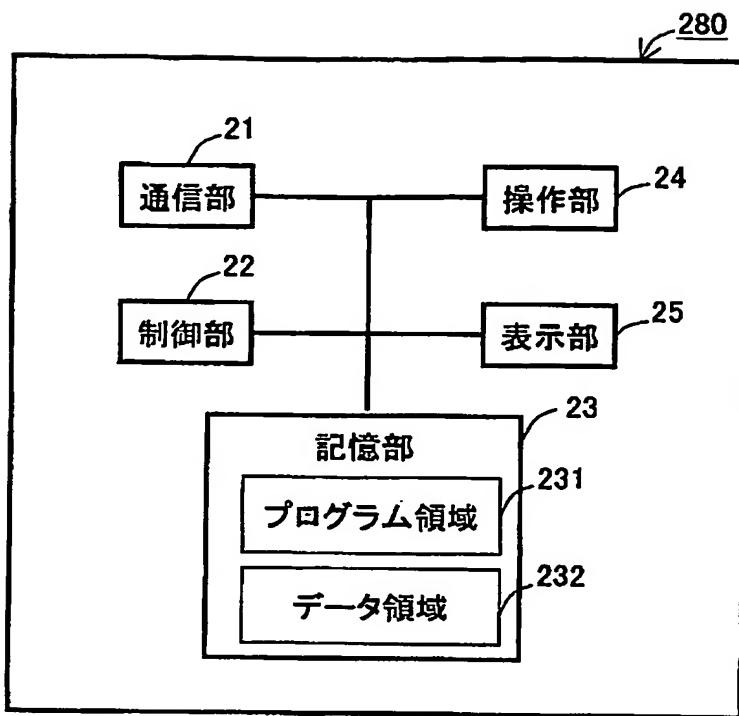
【図38】



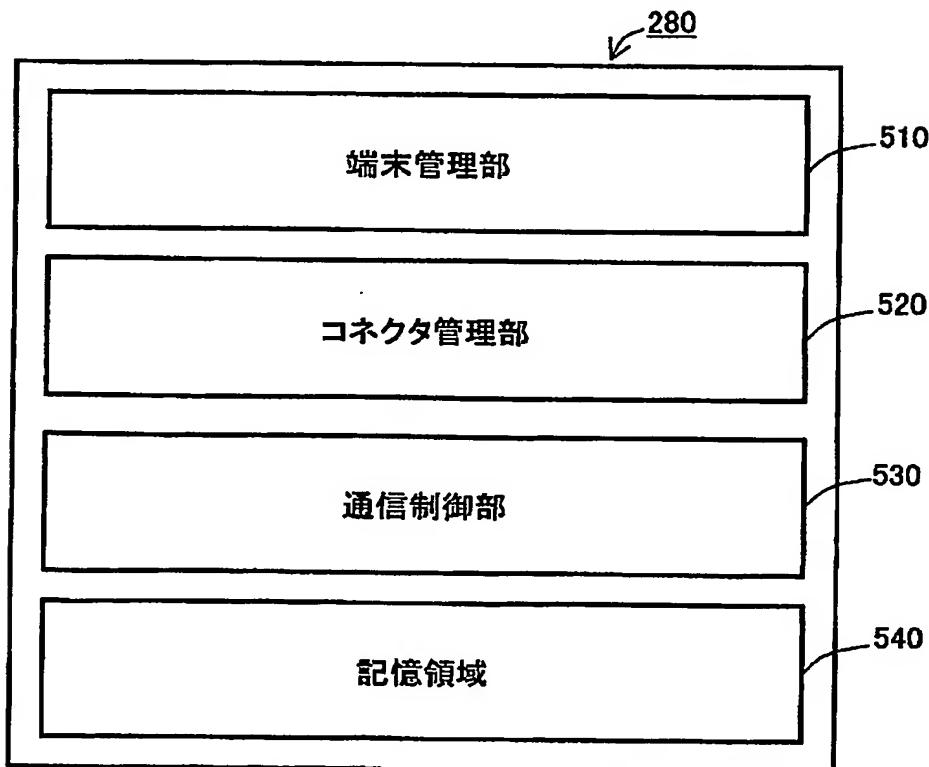
【図39】

一般的なWebサービスの構成

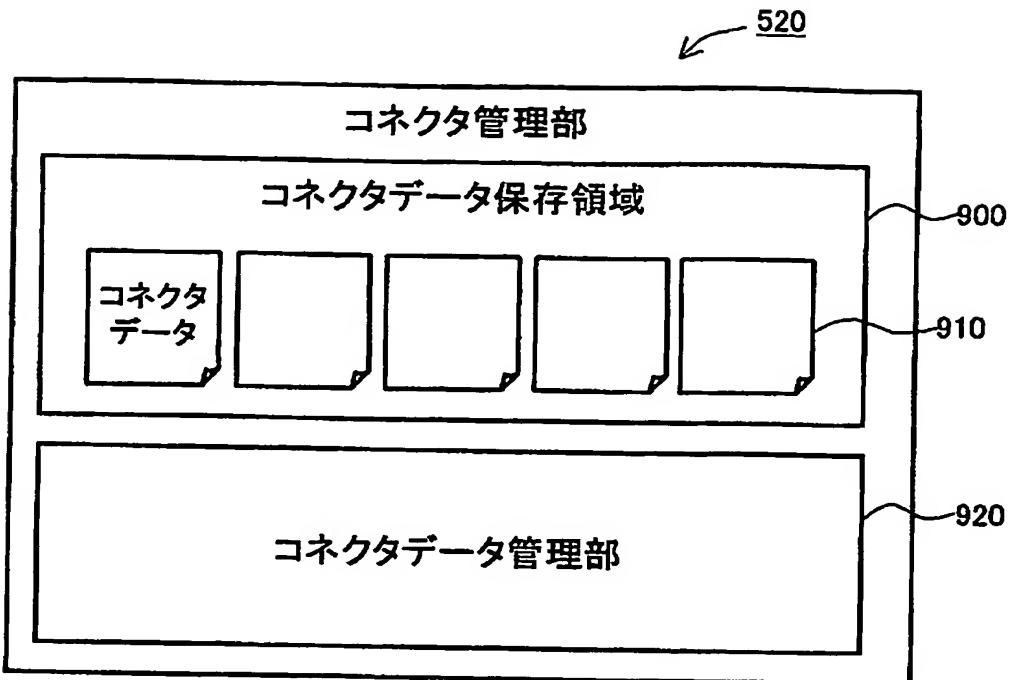
【図40】



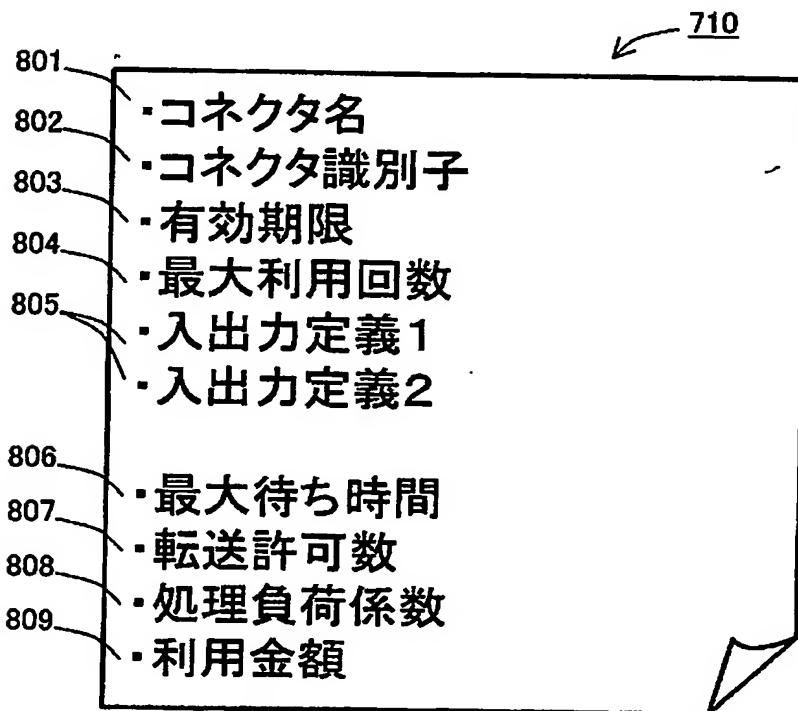
【図41】



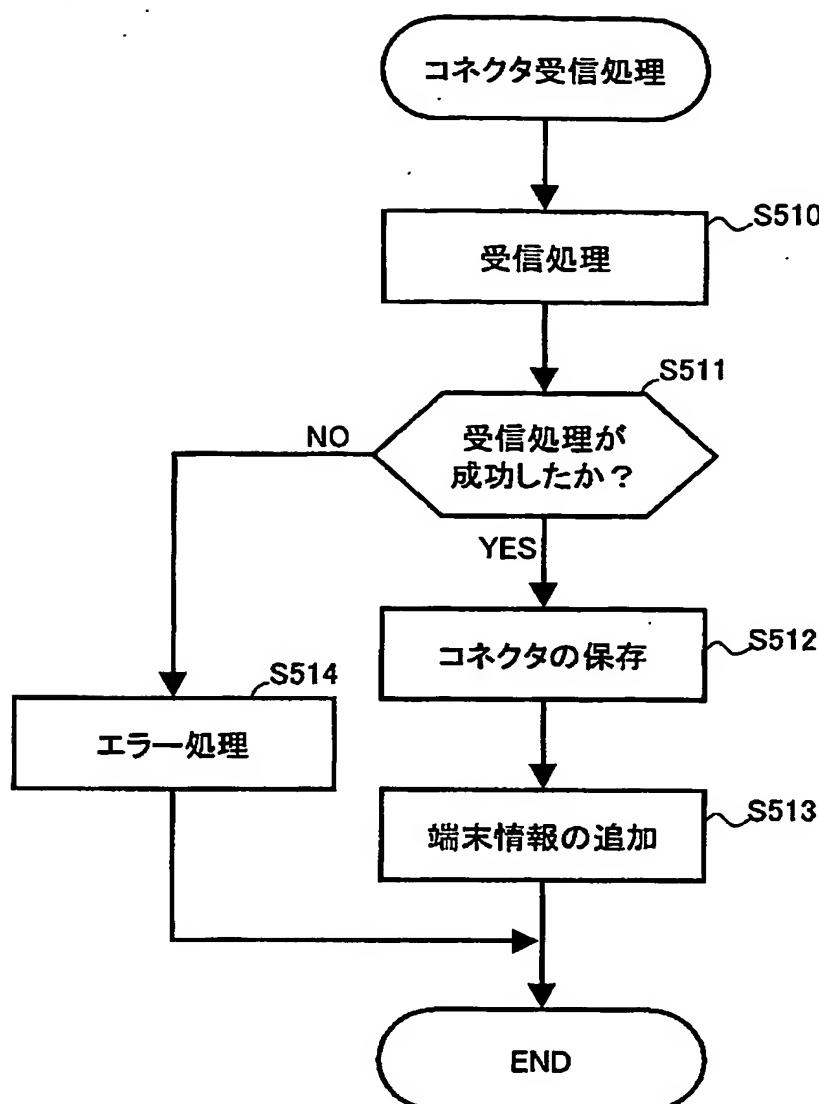
【図42】



【図43】



【図44】



【図45】

730

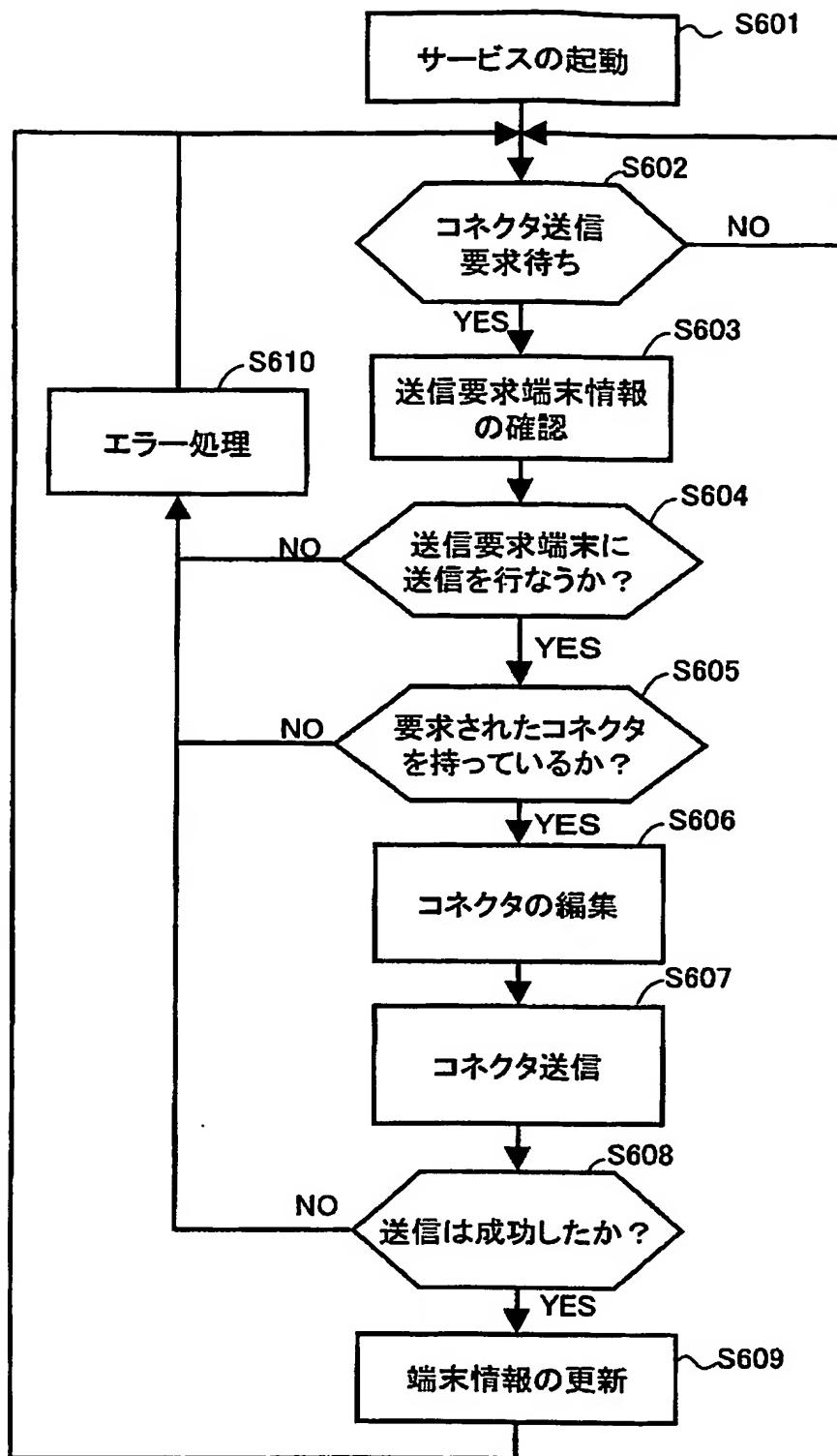
```
<serviceServer>
  <vendor>
    AAA Corp.
  <vendor/>
  <connector>
    <name>
      TextArt
    </name>
    <uniqueNumber>
      LS4369P3550
    </uniqueNumber>
  </connector>
  <registDate>
    2003/4/1
  </registDate>
  <expireNumber>
    100
  </expireNumber>
</serviceServer>
```

【図46】

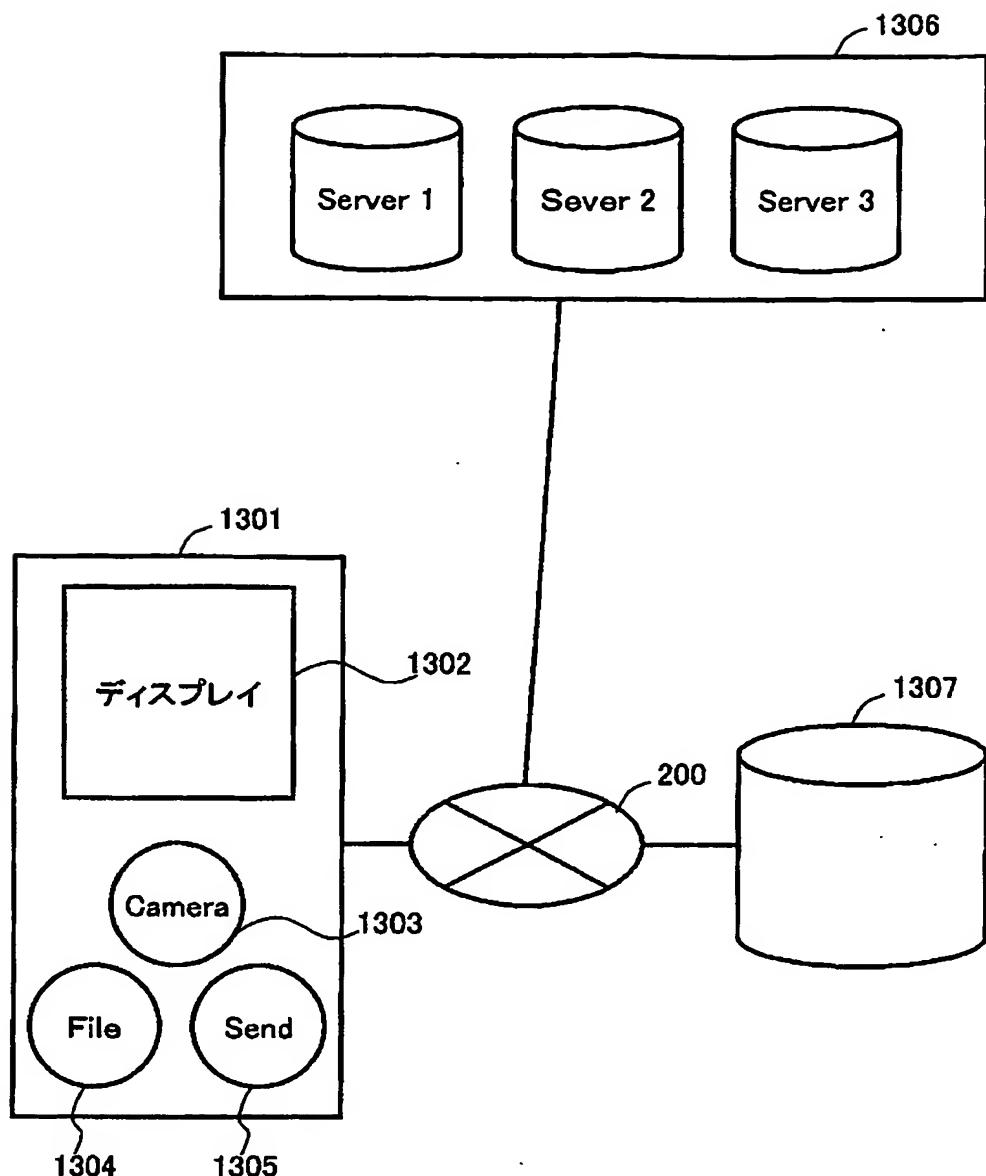
405

```
<smil>
  <head>
    <layout>
      <root-layout width="200" height="200"/>
      <region id="Image" width="150" height="100"
        left="25" top="10" />
      <region id="Text" width="180" height="30"
        left="10" top="120" />
      <button id="TextArt" width="40" height="30"
        left="10" top="160" />
      <button id="Camera" width="40" height="30"
        left="60" top="160" />
      <button id="File" width="30" height="30"
        left="110" top="160" />
      <button id="Send" width="40" height="30"
        left="150" top="160" />
    </layout>
  6023 <action id="TextArt" >
    <setMedia func="name:TextArt" src="Text"
      dest="Text" scope="Server"/>
  </action>
  <action id="Camera" >
    <setMedia func="name:Camera" src=""
      dest="Image" scope="Terminal" />
  </action>
  <action id="File" >
    <setMedia func="name:File" src=""
      dest="Image" scope="Terminal" />
  </action>
  <action id="Send" >
    <setMedia func="name:Send" src=""
      dest="" scope="Terminal" />
  </action>
  </head>
  <body>
    <par>
      <event value="TextArt" action="TextArt" />
      <event value="Camera" action="Camera" />
      <event value="File" action="File" />
      <event value="Send" action="Send" />
    </par>
  </body>
</smil>
```

【図47】



【図48】



【図49】

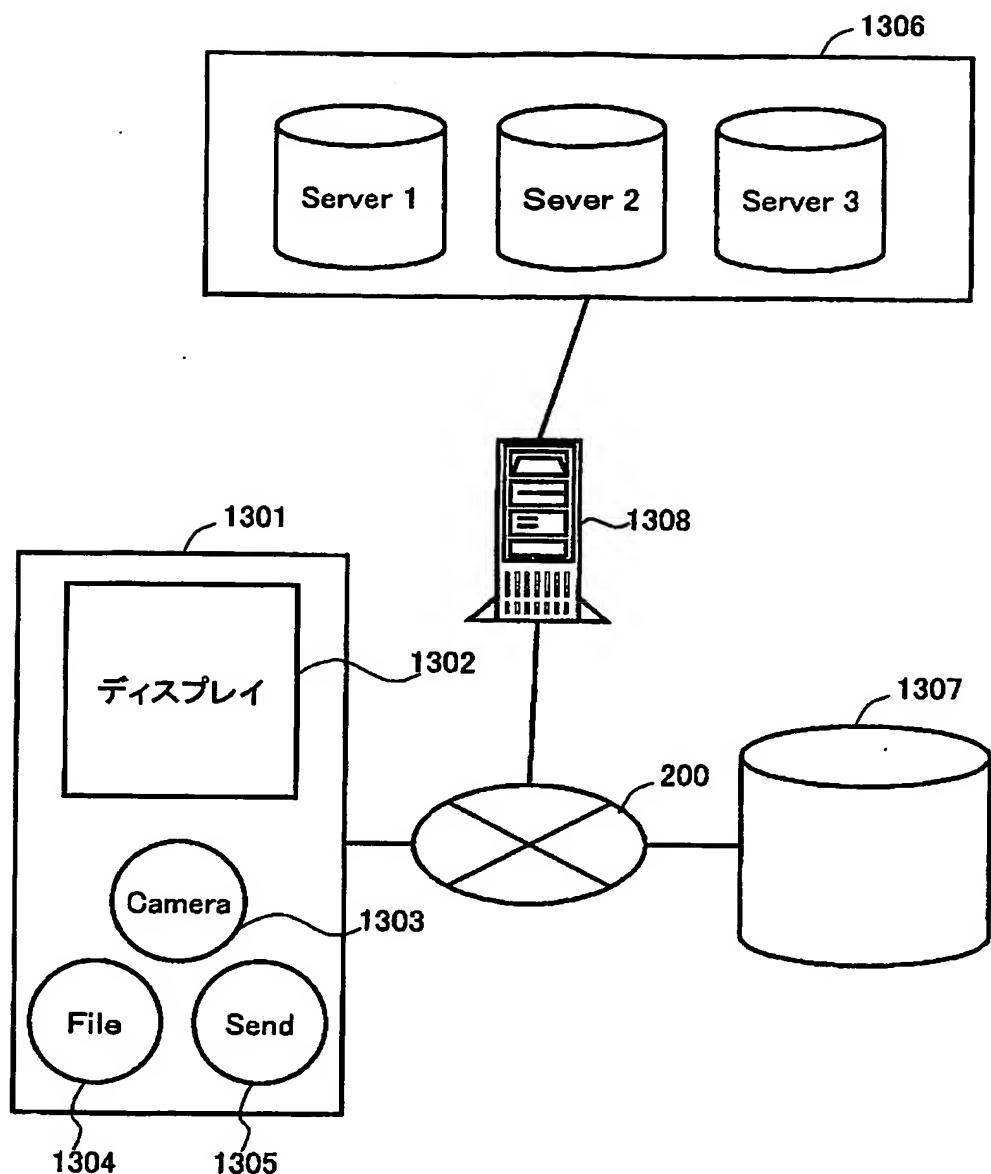
805c

```
<入出力定義>
<型定義 名前="TextArtリクエスト" 形式="文字列" />
<型定義 名前="TextArtレスポンス" 形式="Jpeg画像" />

<機能定義 名前="TextArt">
<入力値 型="TextArtリクエスト" />
<出力値 型="TextArtレスポンス" />
</機能定義>

2703 { <サービス定義 名前="TextArtサービス" />
<アドレス URL="http://virtual1.example.url/TextArt" />
<利用機能 名前="TextArt" />
</サービス定義>
</入出力定義>
```

【図50】



【図51】

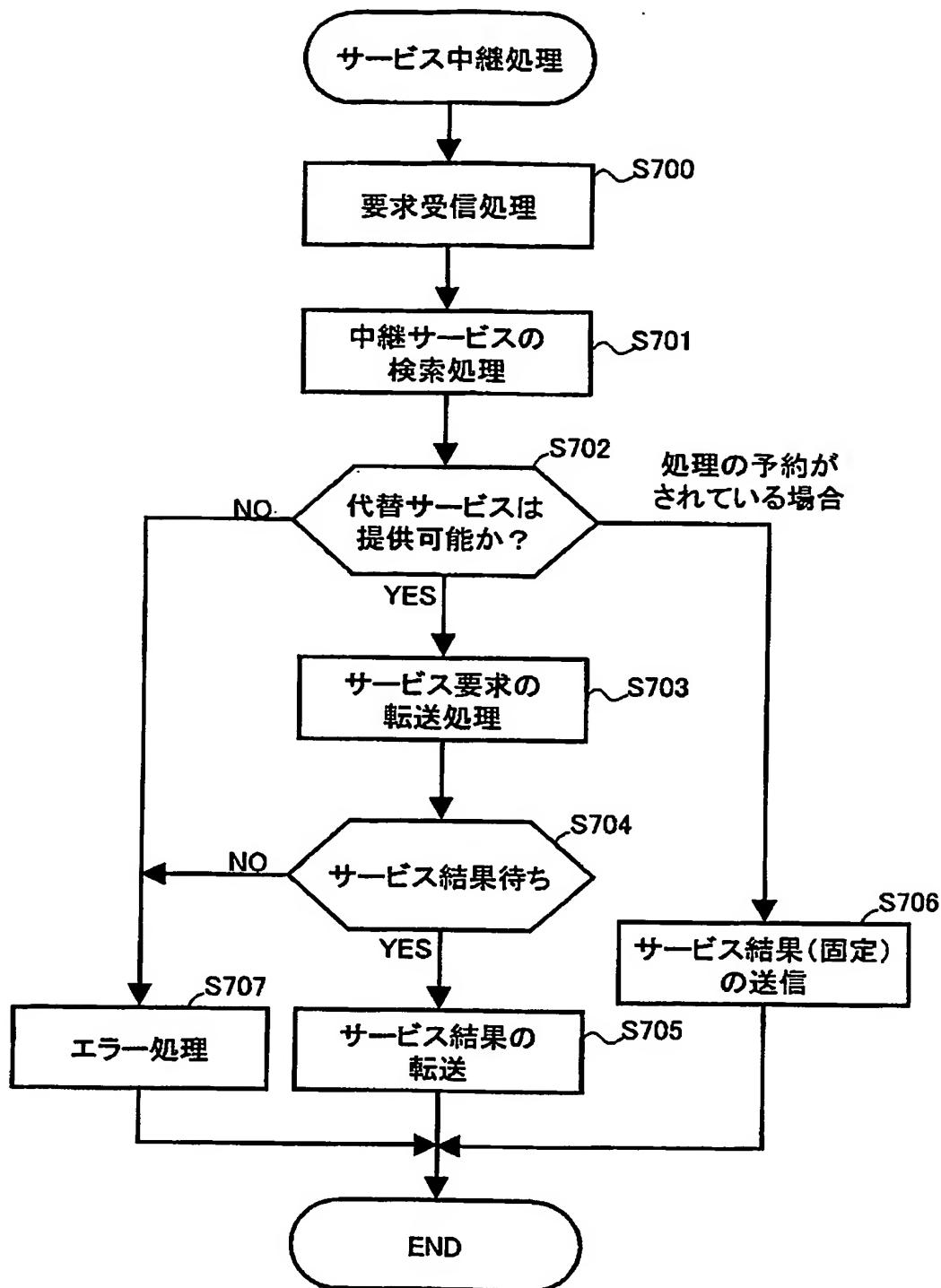
805d

```
<入出力定義>
<型定義 名前="TextArtリクエスト" 形式="文字列" />
<型定義 名前="TextArtレスポンス" 形式="Jpeg画像" />

<機能定義 名前="TextArt">
<入力値 型="TextArtリクエスト" />
<出力値 型="TextArtレスポンス" />
</機能定義>

2803 { <サービス定義 名前="TextArtサービス" />
<アドレス URL="http://connector.server.net/proxy/
example-url/TextArt" />
<利用機能 名前="TextArt" />
</サービス定義>
</入出力定義>
```

【図 52】



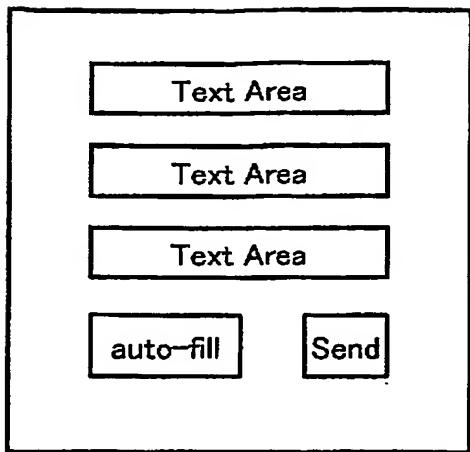
【図 5 3】

405

```
<smil>
  <head>
    <layout>
      <root-layout width="200" height="200"/>
      <region id="Text:1" width="80" height="30"
        left="10" top="10" />
      <region id="Text:2" width="80" height="30"
        left="10" top="50" />
      <button id="Text:3" width="80" height="30"
        left="10" top="90" />
      <button id="auto-fill" width="40" height="30"
        left="10" top="130" />
      <button id="send" width="30" height="30"
        left="60" top="130" />
    </layout>
    <action id="auto-fill" >
      <setMedia func="name:getAddress" src=""
        dest="Text:1" scope="Terminal"/>
      <setMedia func="name:getCompany" src=""
        dest="Text:2" scope="Terminal"/>
      <setMedia func="name:getName" src=""
        dest="Text:3" scope="Terminal"/>
    </action>
  </head>
  <body>
    <par>
      <event value="auto-fill" action="auto-fill" />
      <event value="Send" action="Send" />
    </par>
  </body>
</smil>
```

5301

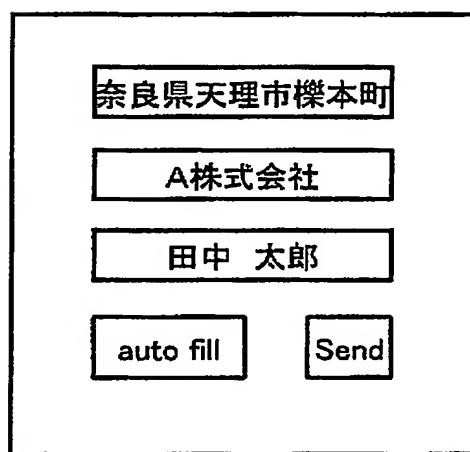
【図54】



Form layout for Figure 54:

- Text Area (three stacked input fields)
- auto-fill (button)
- Send (button)

【図55】



Form layout for Figure 55 with populated fields:

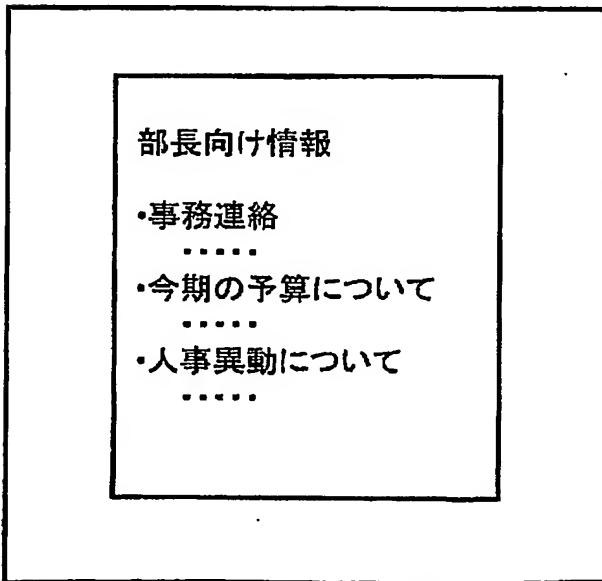
- 奈良県天理市櫟本町 (input field)
- A株式会社 (input field)
- 田中 太郎 (input field)
- auto fill (button)
- Send (button)

【図 5 6】

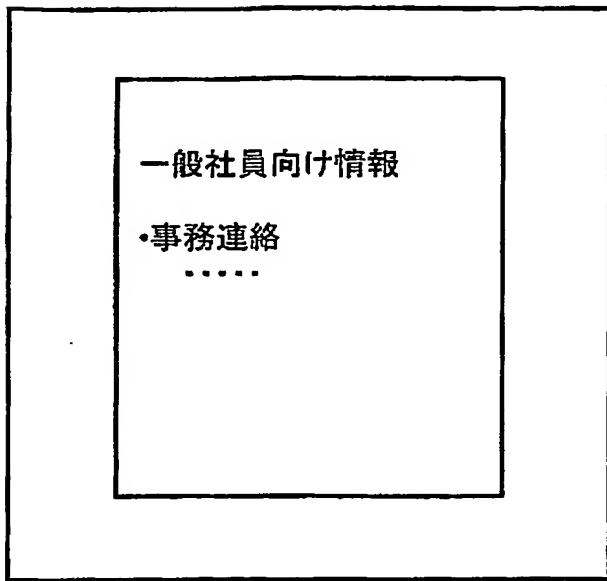
405

```
<smil>
  <head>
    <layout>
      <root-layout width="200" height="200"/>
      <region id="Text" width="80" height="30"
        left="10" top="10" />
    </layout>
  </head>
  <body>
    <par>
      <switch>
        5601 {<text src="secret.txt" region="Text"
          func="name:getRole" scope="Terminal"
          return="chief director" />
        <text src="public.txt" region="Text"
          func="name:getRole" scope="Terminal"
          return="" />
      </switch>
    </par>
  </body>
</smil>
```

【図 5 7】



【図58】

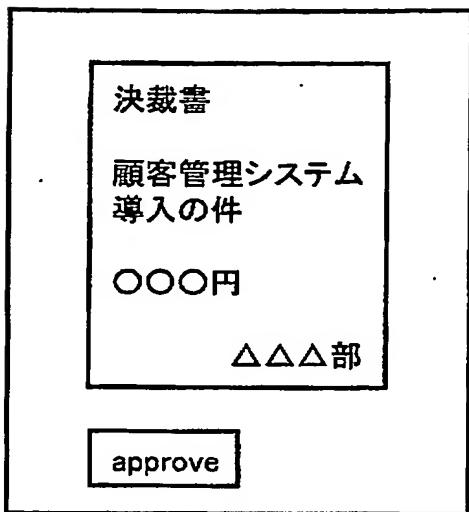


【図59】

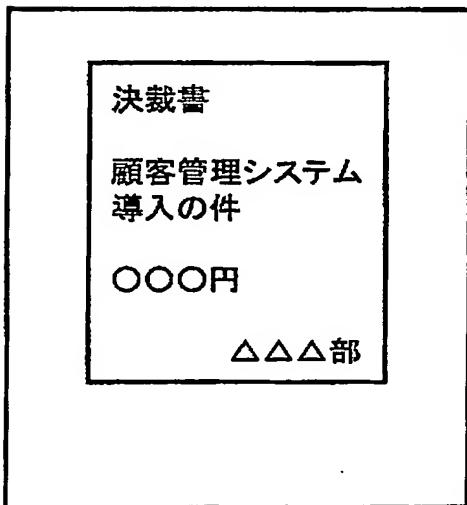
405

```
<smil>
  <head>
    <layout>
      <root-layout width="100" height="200"/>
      <region id="Text" width="80" height="120"
        left="10" top="10" />
      <switch>
        <button id="approve" width="40" height="30"
          left="10" top="140"
          func="name:getRole" return="chief director" >
      </switch>
    </layout>
  </head>
  <body>
    <par>
      <text src="plan.txt" region="Text">
    </par>
  </body>
</smil>
```

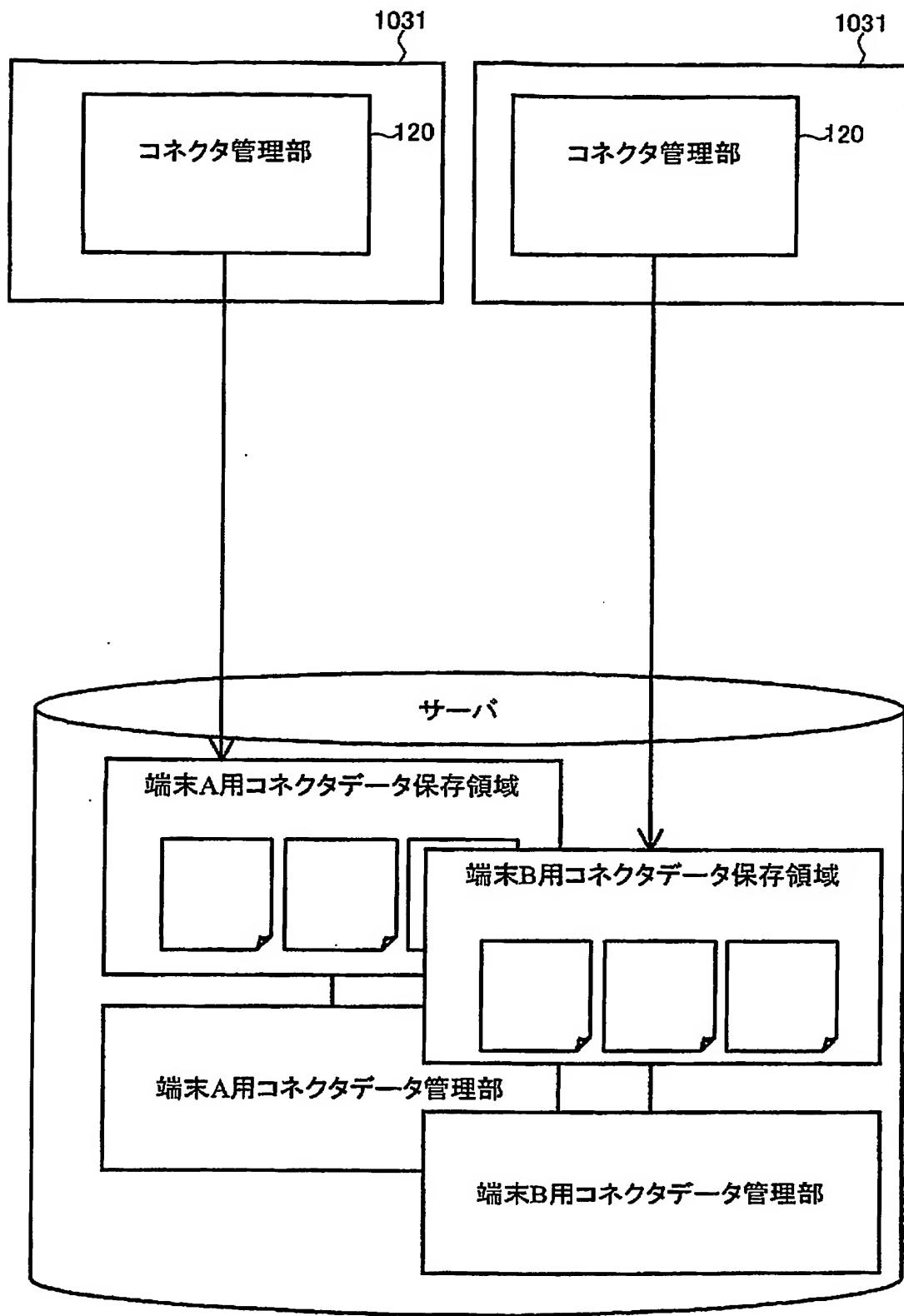
【図60】



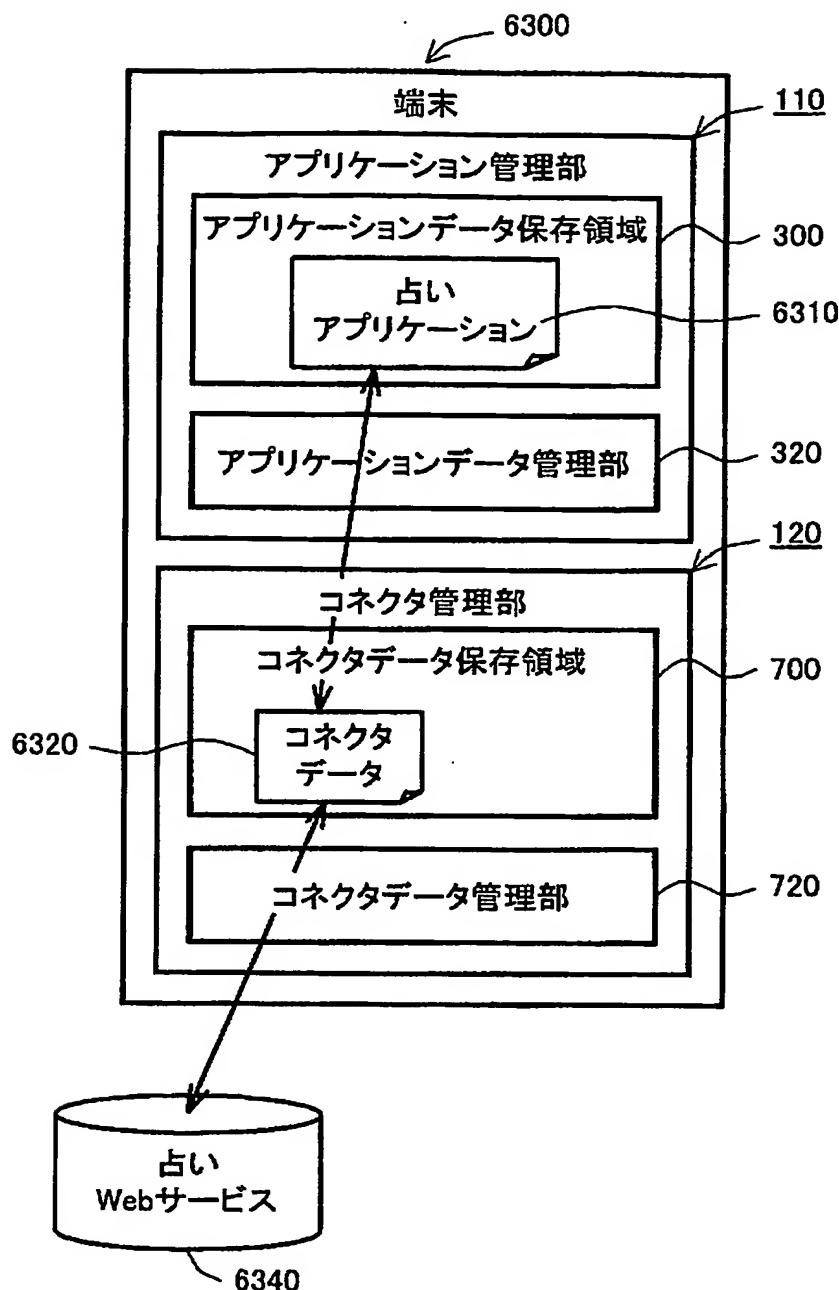
【図61】



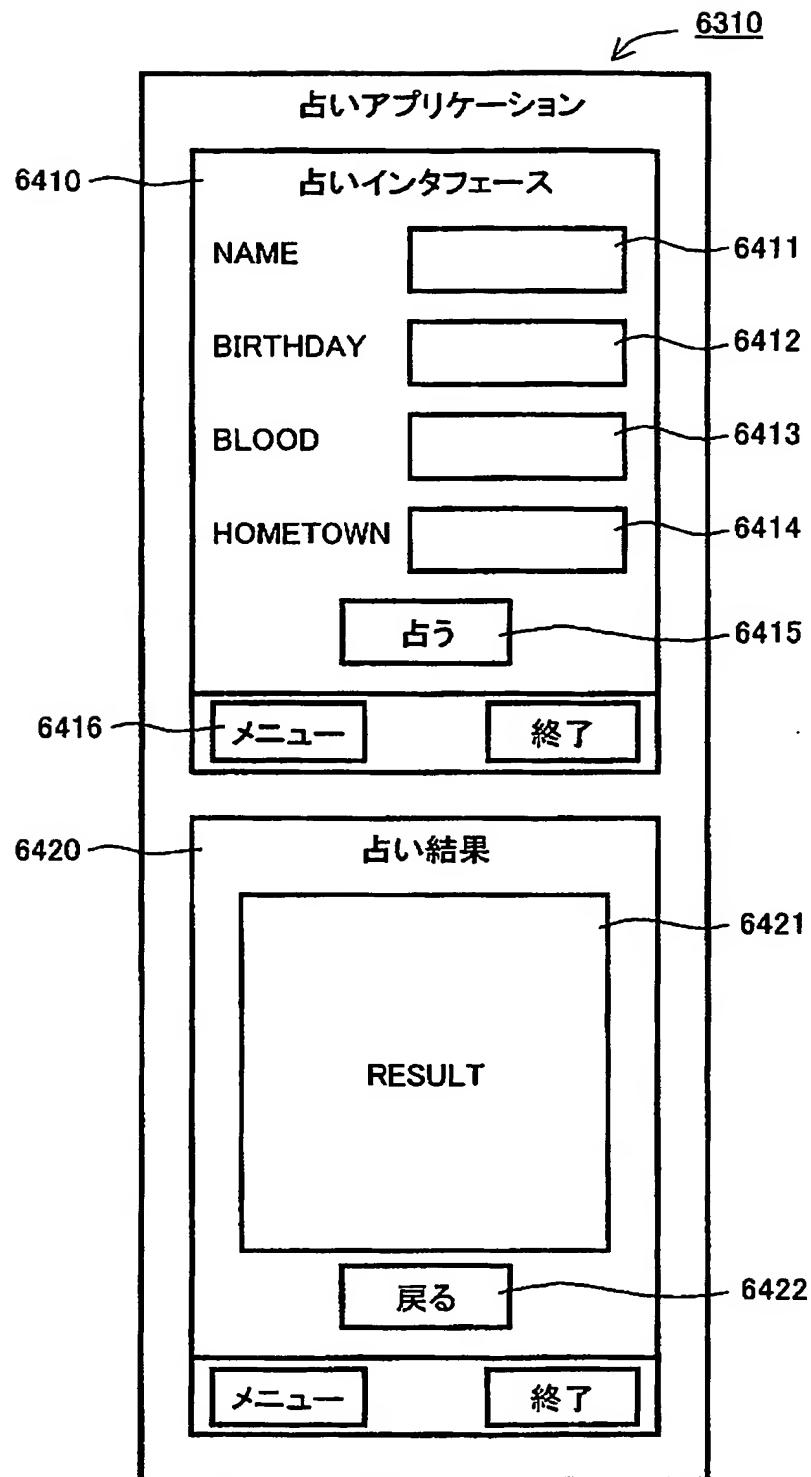
【図 6 2】



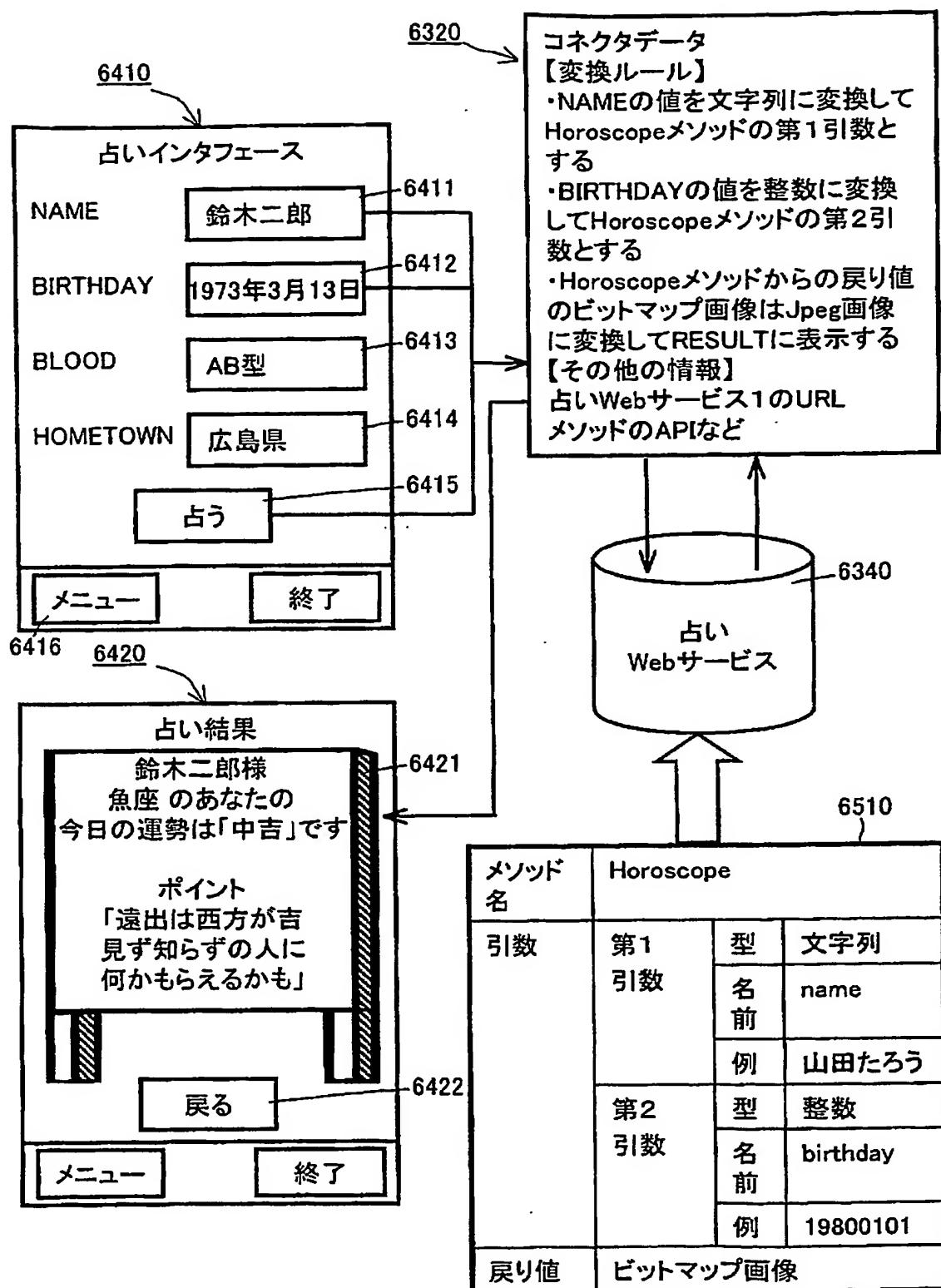
【図63】



【図64】



【図65】



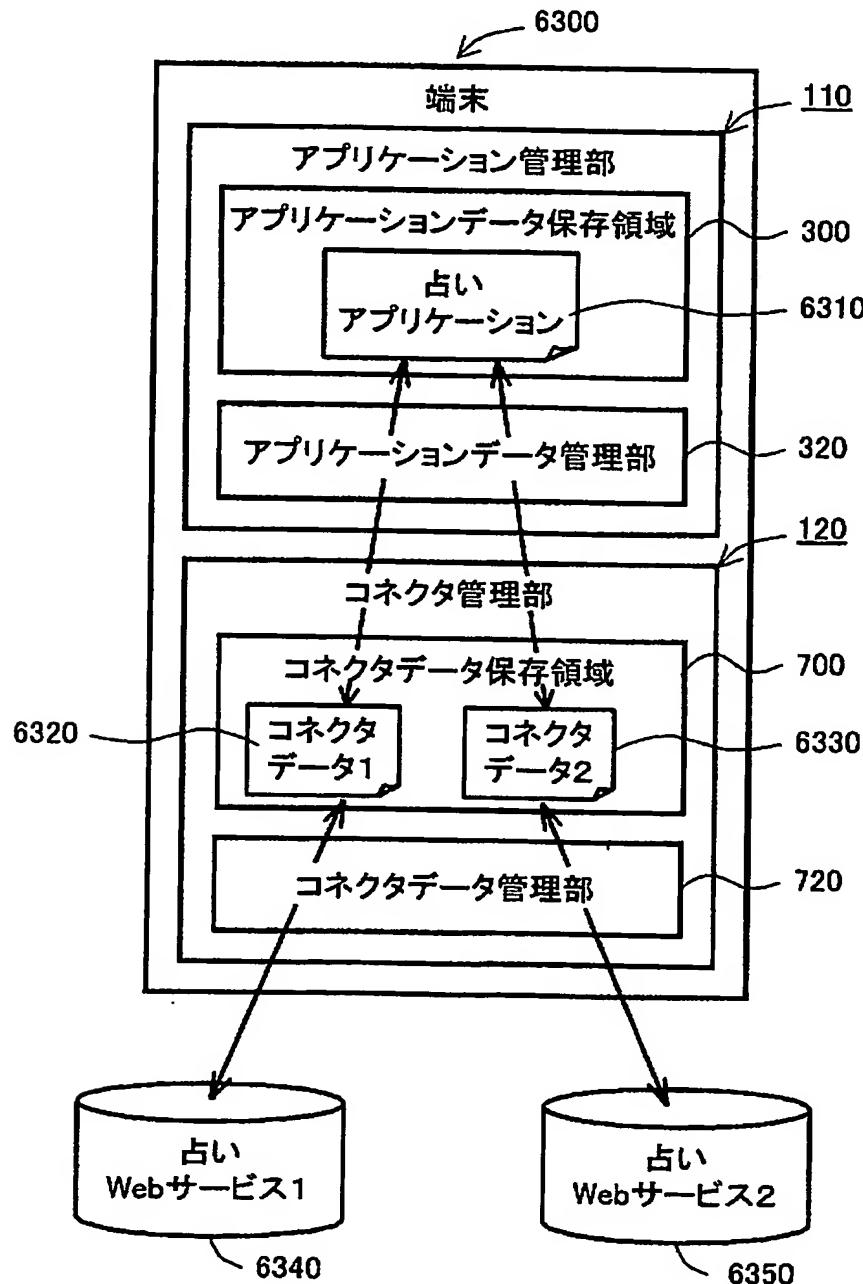
【図66】

```

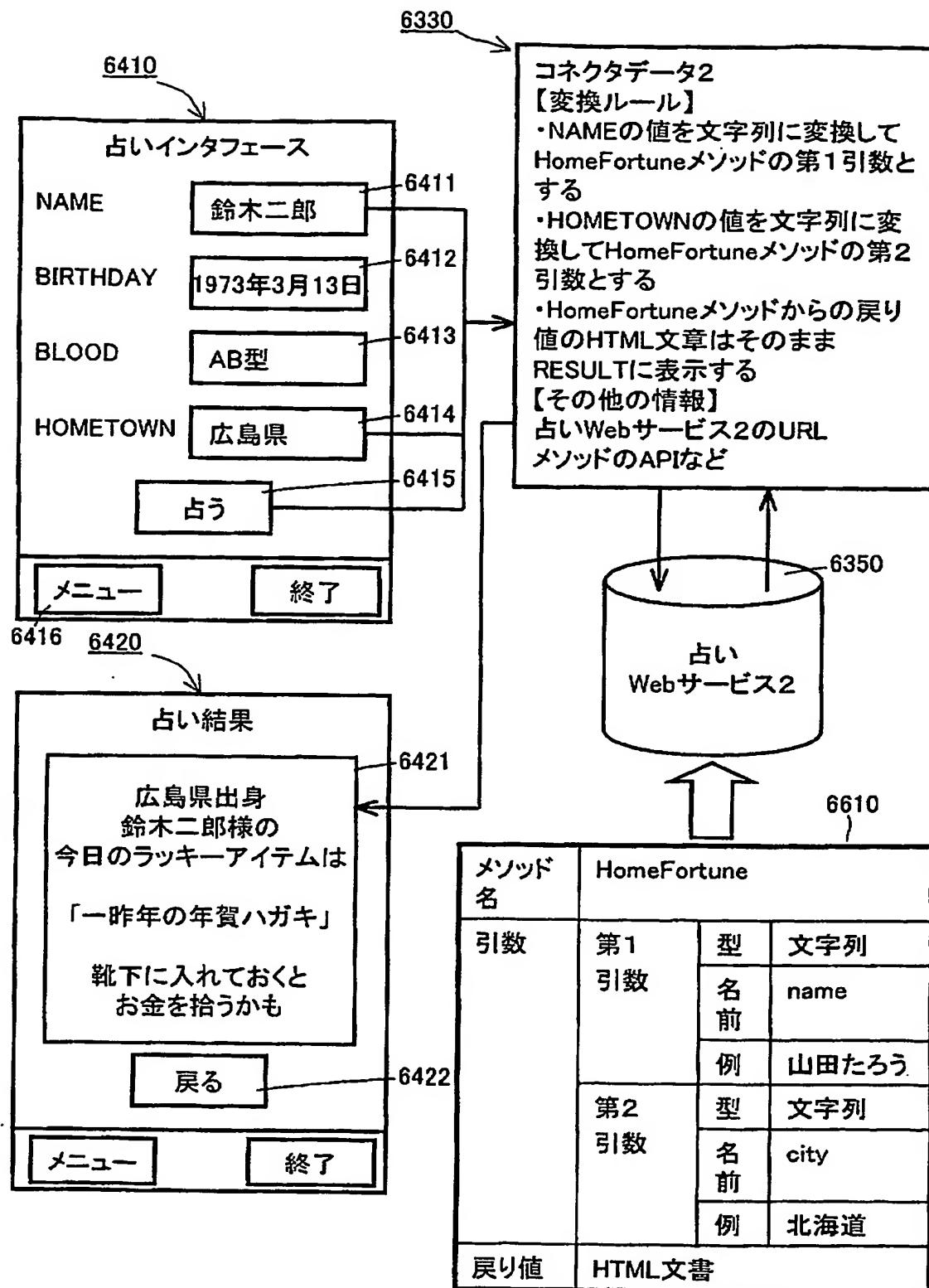
<convert method="Horoscope">
  <paramConv fromApp="NAME" fromType="String"
             toService="name" toType="String" />
  <paramConv fromApp="BIRTHDAY" fromType="String"
             toService="birthday" toType="int" />
  <returnConv fromService="ReturnValue" fromType="bitmap"
              toApp="RESULT" toType="jpeg"/>
</convert>

```

【図67】



【図68】



【図 69】

```

<convert method="HomeFortune">
  <paramConv fromApp="NAME" fromType="String"
             toService="name" toType="String" />
  <paramConv fromApp="HOMETOWN" fromType="String"
             toService="city" toType="String" />
  <returnConv fromService="ReturnValue" fromType="html"
              toApp="RESULT" toType="html"/>
</convert>

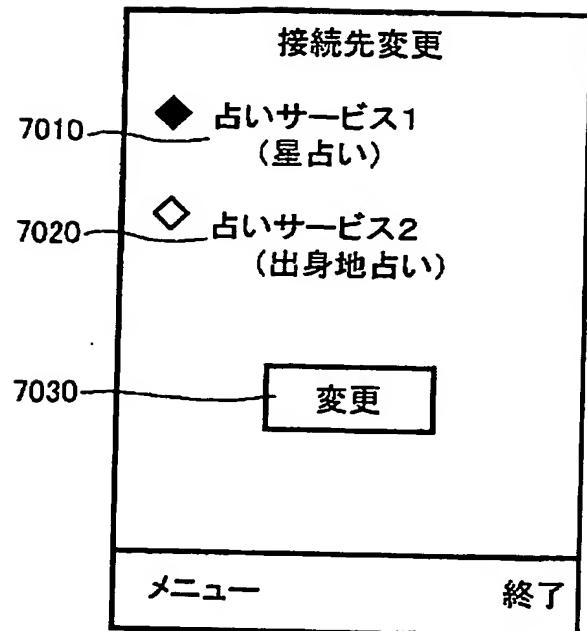
```

【図 70】

6410

占いインターフェース	
NAME	鈴木二郎
BIRTHDAY	1973年3月13日
BLOOD	AB型
HOMETOWN	広島県
6930	接続先変更
6920	アプリケーション変更
6416	メニュー 終了

【図71】



【書類名】要約書

【要約】

【課題】 ネットワーク上で提供される機能を、利便性高く、かつ安全に利用できるサービス利用端末を提供する。

【解決手段】 携帯電話などのサービス利用端末において、ユーザ入力を受付けてアプリケーションを実行する際に、アプリケーションで必要な機能を特定するコネクタを介してアプリケーションに必要な機能を利用する。その際、サービス利用端末は、必要に応じて端末のみがコネクタに記述されているサービス提供サーバに要求して取得する。コネクタを保持するサービス利用アプリケーションの実行に必要な機能を、利便性高く、かつ安全に利用することができる。

【選択図】 図28

特願 2004-071416

出願人履歴情報

識別番号

[000005049]

1. 変更年月日
[変更理由]

1990年 8月29日

新規登録

住所
氏名

大阪府大阪市阿倍野区長池町22番22号
シャープ株式会社